

Oracle[®] Tutor[™]



Section 9 真实应用集群 (Real Application Clusters)

1. Install the Oracle Database 11gR2 software

如果 scan 出错：

验证 scan 的命令 `/u01/app/11.2.0/grid/bin/cluvfy comp scan`

Oracle strongly recommend to use DNS or GNS for SCAN name resolution as hosts file support only one IP for SCAN

If the intention is to use hosts file for SCAN name resolution, and ping command returns correct SCAN VIP, you can ignore the error and move forward.

If the intention is to use DNS or GNS for SCAN name resolution, comment out entries in local hosts file for SCAN name on all nodes, and re-run "`$GRID_HOME/bin/cluvfy comp scan`" to confirm.

```
nslookup scan31.example.com
Server:          192.168.0.90
Address:         192.168.0.90#53
```

```
Name: scan31.example.com
```

Address: 192.168.0.181
Name: scan31.example.com
Address: 192.168.0.131
Name: scan31.example.com
Address: 192.168.0.132

Oracle11gR2Grid Infrastructure (CRS)中引入了 Single Client Access Name (SCAN)来简化客户端对数据库服务的访问，本文将详细说明 SCAN 的含义及其工作原理。

SCAN 概念

Single client access name (SCAN)是提供给所有客户端用于连接集群的虚拟主机名称（与 10g 和 11g R1 中的 vip 主机名称相对应）。

SCAN 是 DNS 或 Grid Naming Service (GNS)中注册到 1-3 个 IP 地址的一个域名。

默认情况下，SCAN 名称与集群的名称相同，且该名称在同一企业内必须唯一。SCAN 的默认值基于本地节点的名称。SCAN 名称的长度最少 1 个字符，最多 15 个字符，必须包含字母和数字，不得以数字开头，可以包含连字符(-)。如果要使用长度超过 15 字符的 SCAN，请在安装 Grid Infrastructure 时选择“Advanced installation”选项。

要想成功安装 Grid Infrasture，SCAN 必须至少解析成一个 IP 地址。

SCAN VIP 地址必须与 virtual IP 地址和 public IP 地址位于同一子网。

Oracle 强烈建议不要再 hosts 文件中配置 SCAN VIP 地址。如果使用 hosts 文件来解析 SCAN 主机名，则该 SCAN hostname 只能解析成一个 SCAN IP 地址。

如果使用 hosts 文件来解析 SCAN 主机名，则在 Grid Infrastructure 安装结束时会遇到类似如下 Cluster Verification Utility 故障。

```
Installation log - $ORACLE_BASE/orainventory/logs/installActions{$TIMESTAMP}.log
INFO: PRVG-1101 : SCANname "cluscan.us.oracle.com" failed to resolve
INFO: ERROR:
INFO: PRVF-4664 : Found inconsistent name resolution entries for SCAN name
"cluscan.us.oracle.com"
INFO: ERROR:
INFO: PRVF-4657 : Name resolution setup check for"scanclunm" (IP address: 10.4.0.202)
failed
OR
PRCS-1016 : Failed to resolve Single Client Access Namecluscan.us.oracle.com
```

出于高可用性和可扩展性的考虑，Oracle 推荐配置使用 DNS 循环解析将 SCAN 解析成三个地址。

因为 SCAN 是与整个集群而不是特定的节点关联，使用 SCAN 可以再无需重新配置客户端的情况下方便地从集群添加或删除节点。SCAN 还增强了数据库的位置无关性，使得客户端配置部再依赖于运行特定数据库的节点。

客户端仍然可以继续使用之前版本的方法来访问集群，不过 Oracle 推荐客户端使用 SCAN 来访问集群。使用 SCAN 的客户端也可以使用 EZCONNECT 方式来访问集群。

Grid Infrastructure 除了启动所有节点上的本地监听程序 LISTENER 来监听本地的 VIP 外，还会启动 SCAN listener LISTENER_SCAN1（一个集群中最多 3 个）来监听 SCAN VIP；Oracle

Database11gR2 默认情况下会将 local_listener 设置成 local LISTENER，将 remote_listener 设置为 SCAN listener。

SCAN listener will be running off GRID_HOME, and by default, in 11gR2 local listener will be running off GRID_HOME as well.

SCAN 相关的常见问题

如何配置 SCAN 和 SCAN listener？

在“Typical installation”过程中，程序会提示您确认默认的 Single Client Access Name (SCAN)。如果您更改默认的 SCAN，则您要使用的 SCAN 名称必须保持唯一性。

如果 SCAN 名称解析成一个 IP 地址，则 root 脚本(root.sh or rootupgrade.sh)将根据 SCAN 名称解析的 IP 地址数量来创建 SCAN VIP 资源(ora.scan1.vip)以及相应 SCAN listener 资源(ora.LISTENER_SCAN1.lsnr)的数量。例如，如果 SCAN 名称可以解析为 2 个 IP 地址，则将 root 脚本就会创建两个 SCAN VIP 资源和两个相应的 SCAN listener 资源。

SCAN VIP 及相应的 SCAN listener 成对工作，当 SCAN VIP failover 到其他节点，则相应的 SCAN 监听程序也会 failover 到同一节点。

当 SCAN VIP failover 发生时，始终会选择运行 SCAN VIP 最少的节点，例如，在一个 4 节点的集群中，如果 SCAN VIP 运行在节点 1、节点 2 和节点 3 上，则当节点 3 down 掉时，节点 3 上的 SCAN VIP 及相应的 SCAN listener 就会 failover 到节点 4 上，因为其他两个节点的每个节点上已经运行了一个 SCAN VIP。

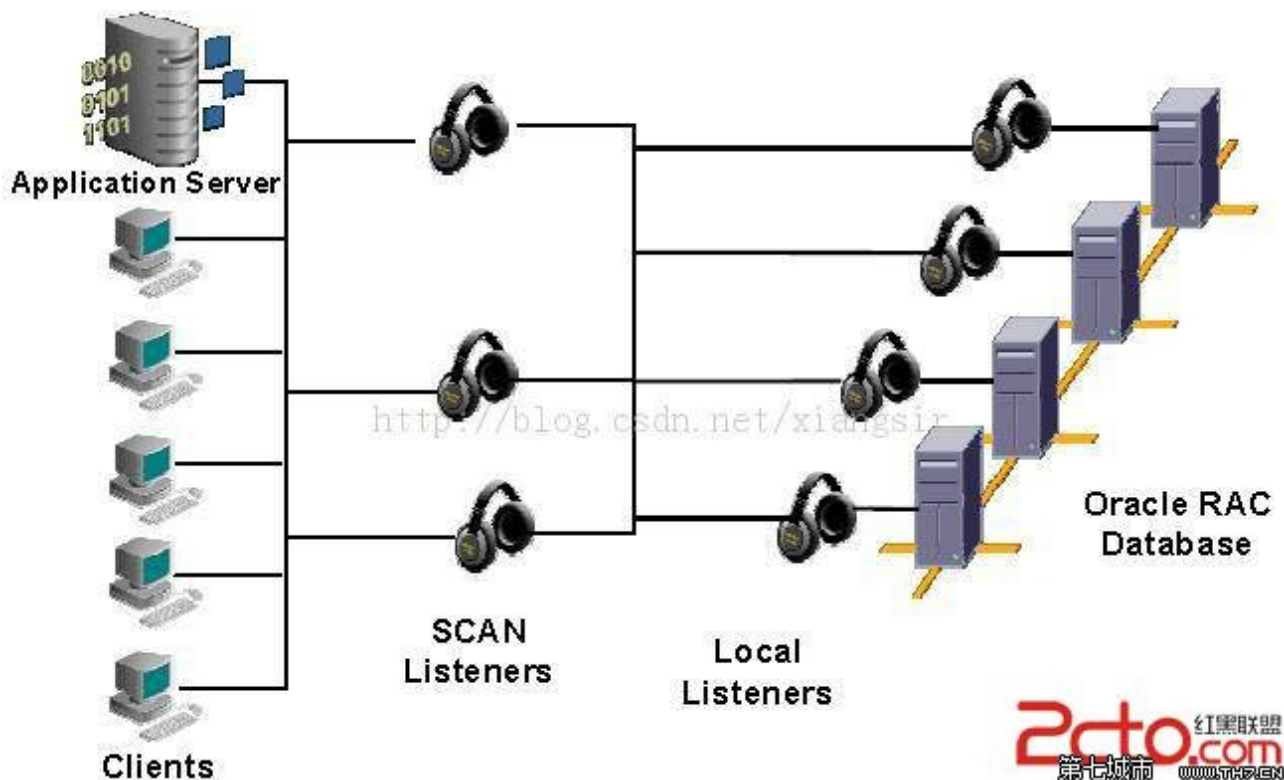
我们还可以使用“srvctl”来添加/修改 SCAN vip 和 listener。

是否仍然需要在每个节点上配置 local_listener？

回答是“是”，我们仍然需要在每个节点配置独立的 local listener。SCAN listener 并没有取代每个节点上的 local listener。

在 11g R2 的 GI 中，集群中的所有节点上会运行 3 个新的 scan listener，如果集群节点超过 3 个，则只会在其中 3 个节点上运行 SCAN listener，无论集群有多少节点，最多也只会 3 个 scan listener。数据库通过 remote_listener 初始化参数注册到 SCAN listener 中。如果集群中任意一个 scan listener 挂掉，都会在一个新的节点上自动重启。

SCAN 的工作原理：



在客户端提交请求时，在 SCAN VIP 地址及端口上进行监听的 SCANlistener 就会代表客户端建立连接。因为集群上的所有服务都已注册到 SCAN listener 中，SCAN listener 就会将负载最少的节点（每个 scan listener 会实时更新集群的负载统计信息）上的 local listener 的地址返回给客户端。最后，客户端就与通过返回的提供服务的节点上的 local listener 与服务建立连接。所有这些操作对于客户端来说都会透明的，客户端无需进行任何明确的配置。

```
$ SRVCTL STATUSSCAN_LISTENER
SCAN ListenerLISTENER_SCAN1 is enabled
SCAN listenerLISTENER_SCAN1 is running on node TEST1
SCAN ListenerLISTENER_SCAN2 is enabled
SCAN listenerLISTENER_SCAN2 is running on node TEST2
SCAN ListenerLISTENER_SCAN3 is enabled
SCAN listenerLISTENER_SCAN3 is running on node TEST3
```

可否用 '/etc/hosts'来解析 SCAN 而不用 DNS 或 GNS？
Oracle 强烈建议不要在 hosts 文件中配置 SCAN VIP 地址。但是可以使用 hosts 文件来解析 SCAN 名称，只不过只能解析成一个 SCAN IP 地址。

如果使用 hosts 文件来解析 SCAN 名称，则在安装结束时会遇到 ClusterVerification Utility 失败错误，详细信息请见 metalink 文章 NOTE 887471.1。

可否使用之前的连接方式（VIP）来进行客户端连接？
客户端仍旧可以使用之前版本的连接方式（VIP）来访问集群。
Oracle 强烈建议客户端使用 SCAN 来访问集群。使用 SCAN 的客户端也可以 EZCONNECT 方式来访问集群。

是否必须要使用 SCAN ?
强烈建议而已，并非必须。

是否支持删除 SCAN ?
SCAN Oracle 11gR2 Grid Infrastructure 的一个组件，不支持删除 SCAN。

Sample TNS entry for SCAN

```
TEST.Oracle.COM =  
(DESCRIPTION=  
(ADDRESS=(PROTOCOL=tcp)(HOST=SCAN-TEST.Oracle.COM)(PORT=1521))  
(CONNECT_DATA=(SERVICE_NAME=11GR2TEST.Oracle.COM))  
)
```

Sample TNS Entry without SCAN

```
TEST.Oracle.COM =  
(DESCRIPTION=  
(ADDRESS_LIST=  
(ADDRESS=(PROTOCOL=tcp)(HOST=TEST1-vip.Oracle.COM)(PORT=1521))  
(ADDRESS=(PROTOCOL=tcp)(HOST=TEST2-vip.Oracle.COM)(PORT=1521))  
)  
(CONNECT_DATA=(SERVICE_NAME=11GR2TEST.Oracle.COM))  
)
```

如果 crsd 出错：

crsctl check cssd

crsctl check cluster -n station31

crsctl query css votedisk

2.Configure ASM for the shared disks and create a clustered database

In our case, as we have the disks and disk OCR voting in ASM, we must make a stop arranged crsctl (Metalink note 984663.1).

Oracle 11gR2 RAC 默认情况下 gsd 服务是禁用的，但某些客户觉得 OFFLINE 看起来不爽，下面贴出一篇 metalink 文章，解释 gsd 服务的作用：GSD Is Used Only if 9i RAC Database is Present (文档 ID 429966.1)

GSD Is Used Only if 9i RAC Database is Present (文档 ID 429966.1)

修改时间:2013-3-11 类型:REFERENCE

In this Document

[Purpose](#)

[Scope](#)

[Details](#)

[Scalability RAC Community](#)

[References](#)

APPLIES TO:

Oracle Database - Enterprise Edition - Version 10.1.0.2 to 11.2.0.0 [Release 10.1 to 11.2]

Information in this document applies to any platform.

Checked for relevance on 11-Mar-2013

PURPOSE

The purpose of this document is to clarify the functionality of GSD in 10g and above.

SCOPE

This article is informational and intended for any user.

DETAILS

The function of GSD (10g and above) is to service requests for 9i RAC management clients and therefore when there are no 9i databases present, there is nothing for GSD to do.

Consequently, there will be no impact on a RAC cluster if GSD is offline and 9i is not used.

If gsd fails to start due to whatever reasons then best thing is to work with Oracle support to analyze and fix the issue. Until that time, gsd can be temporarily disabled.

In 11.2 GSD is disabled by default and the service will show as target:offline, status:offline.

Disable GSD (pre 11.2)

After confirming that there are no 9i databases being used you can disable GSD by adding 'exit 0' after the initial comments in the script \$ORACLE_HOME/bin/gsdctl where \$ORACLE_HOME is the home from which nodeapps are running (i.e. crs home).

```
#case $ORACLE_HOME in
# "") echo "****ORACLE_HOME environment variable not set!"
```

```
# echo " ORACLE_HOME should be set to the main"
# echo " directory that contains Oracle products."
# echo " Set and export ORACLE_HOME, then re-run."
# exit 1;;
#esac
exit 0 ## Manually added as a temporary workaround
```

A backup of the original script should be made before making the above change.

Disable GSD (11.2)

You may want to disable GSD after you upgraded all your Oracle9i RAC databases.

```
srvctl stop nodeapps
srvctl disable nodeapps -g
srvctl start nodeapps
```

Enable GSD in 11.2

```
srvctl enable nodeapps -g
srvctl start nodeapps
```

see also:

Grid Infrastructure Installation Guide

5 Oracle Grid Infrastructure Postinstallation Procedures

5.3.3 Enabling The Global Services Daemon (GSD) for Oracle Database Release 9.2

以下做法适用于 11.2.0.1 不适用于 11.2.0.2、11.2.0.3 和 11.2.0.4。在 11g R2 中，默认 oc4j 和 gsd 资源是 disable 的；oc4j 是用于 WLM 的一个资源,WLM 在 11.2.0.2 才可用；gsd 是 CRS 用于跟 9iRAC 进行通信的一个模块,是为了向后兼容才保留的，不影响性能；建议不要删除,也不要尝试开启他们,忽略即可。

1：修复 GSD 服务：

```
srvctl enable nodeapps -g
srvctl start nodeapps
```

2：修复 oc4j 服务：

```
srvctl enable oc4j
srvctl start oc4j
```



```
[grid@12345-node02 ~]$ crs_stat -t
```

| Name | Type | Target | State | Host |
|----------------|----------------|---------|---------|-------------|
| ----- | | | | |
| ora....SM1.asm | application | ONLINE | ONLINE | 1234...de01 |
| ora....01.lsnr | application | ONLINE | ONLINE | 1234...de01 |
| ora....e01.gsd | application | OFFLINE | OFFLINE | |
| ora....e01.ons | application | ONLINE | ONLINE | 1234...de01 |
| ora....e01.vip | ora....t1.type | ONLINE | ONLINE | 1234...de01 |
| ora....SM2.asm | application | ONLINE | ONLINE | 1234...de02 |
| ora....02.lsnr | application | ONLINE | ONLINE | 1234...de02 |
| ora....e02.gsd | application | OFFLINE | OFFLINE | |
| ora....e02.ons | application | ONLINE | ONLINE | 1234...de02 |
| ora....e02.vip | ora....t1.type | ONLINE | ONLINE | 1234...de02 |
| ora.DATA.dg | ora....up.type | ONLINE | ONLINE | 1234...de01 |
| ora....ER.lsnr | ora....er.type | ONLINE | ONLINE | 1234...de01 |
| ora....N1.lsnr | ora....er.type | ONLINE | ONLINE | 1234...de01 |
| ora.asm | ora.asm.type | ONLINE | ONLINE | 1234...de01 |
| ora.cvu | ora.cvu.type | ONLINE | ONLINE | 1234...de01 |
| ora.qsd | ora.qsd.type | OFFLINE | OFFLINE | |
| ora....network | ora....rk.type | ONLINE | ONLINE | 1234...de01 |
| ora.oc4j | ora.oc4j.type | ONLINE | ONLINE | 1234...de01 |
| ora.ons | ora.ons.type | ONLINE | ONLINE | 1234...de01 |
| ora.scan1.vip | ora....ip.type | ONLINE | ONLINE | 1234...de01 |

3.Configure archiving

4.Configure services using both Manual and Policy Managed methods

```
srvctl config database -d racdb
```

Database unique name: racdb

Database name: racdb

Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1

Oracle user: oracle

Spfile: +DATA/racdb/spfileracdb.ora

Domain:

Start options: open

Stop options: immediate

Database role: PRIMARY

Management policy: AUTOMATIC

Server pools: srvpool,servpool2

Database instances:

Disk Groups: DATA,FRA

Mount point paths:

Services: serv1

Type: RAC

Database is policy managed

```
srvctl modify database -d racdb -g servpool3
```

```
srvctl config srvpool -g servpool3
```

数据库名称: racdb

RAC 集群两节点名称 : racdb01 , racdb02

Policy-Managed 方式介绍

基于策略的管理方式，是以服务器池（Server Pools）为基础的，简单地说，就是先定义一些服务器池，池中包含一定量的服务器，然后再定义一些策略，根据这些策略 Oracle 会自动决定让多少数据库实例运行在池中的几台机器上。数据库实例名后缀、数据库实例个数、所运行的主机，这些都是通过策略决定的，而不是数据库管理员事先定好的。

何种环境适合使用这种新的方式进行管理？

当管理大量的服务器集群，并且在这些集群中运行着多种不同重要程度，不同策略的 RAC 数据库时，为了简化管理，建议使用 Policy- Managed 方式，实际上 Oracle 也建议只有在超过 3 台的服务器的时候才使用 Policy-Managed 来管理整个数据库集群。想象一下使用 Policy-Managed 方式可以达到的效果：如果我们有 10 台服务器组成，根据不同的应用的重要性定义服务器池的关键程度，然后在其中某些机器意外 停机的情况下，仍然可以自动地保持足够多的机器给重要的系统提供数据库服务，而将不关键的系统数据库服务器个数降低到最低限度。

策略管理：DBA 指定数据库资源运行在哪个服务器池(排除 generic or free)。Oracle Clusterware 负责将数据库资源放在一台服务器。

Policy managed: Database administrators specify in which server pool (excluding generic or free) the database resource will run. Oracle Clusterware is responsible for placing the database resource on a server.

服务器以如下次序被分配入服务器池：

Generic server pool

User assigned server pool

Free

Oracle Clusterware 使用服务器池的重要性决定分配服务器次序：

1. 按重要性次序分配服务器给所有服务器池，直到满足服务器池的最小数目要求
2. 按重要性次序分配服务器给服务器池，直到它们满足服务器池的最大数目要求
3. 默认，任何剩下的服务器加入 FREE 服务器池

策略管理数据库背后的目标是删除到 1 个特定实例或服务 服务的硬编码

数据库可以和 1 个服务器池关联(而不是特定的节点集)。服务器池决定被资源(数据库，服务，第三方应用程序)所需的最小和最大服务器数目。

数据库实例将运行在已被分配给服务器池的服务器上。(使用 min_size 决定数据库必需运行在哪些服务器，以及必需运行在多少服务器上)

既然被分配给服务器池的服务器可以动态地变更，这允许 Oracle 基于集群可用的服务器总数动态地交付服务。

数据库实例将启动在足够多的服务器上(受制于服务器的可用性)。无需硬编码规定数据库实例运行在哪些服务器上。

数据库的任何实例可以运行在任何节点上。在实例号和节点之间无固定的映射关系。

当服务器被释放/添加/删除时，他们按之前提及的规则被分配到存在的服务器池中。

理论上的例子

例如，如果 1 个集群，总共有 8 个节点组成，并且支持 3 个 RAC 数据库。每个数据库将定义服务器的最小和最大数目。

假设 DB1 定义最小 4 台、最多 6 台服务器(重要性为 10)，

假设 DB2 定义最小 2 台、最多 3 台服务器(重要性为 7)，

假设 DB3 定义最小 2 台、最多 3 台服务器(重要性为 5)。

初始 8 节点将被配置成节点 1-4 被分配给 DB1，节点 5-6 被分配给 DB2，节点 7-8 被分配给 DB3。如果节点 3 由于某种原因发生故障，系统将分配节点 7 或 8 给 DB1，因为其比 DB3 有更高的重要性而且最小需要 4 台服务器，即使将导致 DB3 降到最小服务器水平以下。

如果节点 3 被重新激活，将被立即分配给 DB3 以使数据库恢复到最小所需的服务器数。

如果第 9 个节点被添加到集群，将被分配给 DB1，因为其重要性最高而且未满足最大服务器数。

Admin-Managed 方式介绍

实际上上面的表述已经明确说明了，Policy-Managed 和 Admin-Managed 方式的差别。

让我们再回顾一下，在以往我们创建一个 RAC 数据库大概是怎样的方法，我们在 dbca 的界面中会选择要将数据库实例运行在整个集群中的几台机器上，或者是 2 台或者是 3 台，甚或是更多，但是只要在安装的时候选定几台机器，那么以后如果不做增减节点的操作，就始终会在这几台机器上运行。而且，通常会根据主机名称的排序自动将每台主机上的数据库实例依次命名为 dbname1 到 dbnameN。这些在管理员安装完毕以后，都不会再自动变化，这就是 Admin-Managed 方式。

管理员管理：DBA 指定数据库资源运行的所有服务器，并且按需手动放置资源。这是之前版本 Oracle 数据库使用的管理策略。

Administrator-managed: Database administrators define the servers on which databases resource run, and place resources manually as needed. This is the management strategy used in previous releases.

Policy-Managed 方式和 Admin-Managed 方式 Service 使用例子

1, 如何检查是否 Admin managed 方式:

```
[grid@racdb01 ~]$ srvctl config database -d racdb
Database unique name: racdb
Database name: racdb
Oracle home: /app/product/oracle/11.2.0.4/db_1
Oracle user: oracle
Spfile: /app/product/oracle/11.2.0/db_1/dbs/spfileracdb1.ora
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools: racdb
Database instances: racdb1,racdb2
Disk Groups: ASM_DATA,ASM_FRA
Mount point paths:
Services: rac_first,rac_second
Type: RAC
Database is administrator managed
```

这就说明是以 Admin Managed 来管理 RAC。

2, 下面举例子来说明如何检查和修改数据库服务器池及 Service

A, 添加服务器池 mypool(最小数目 0, 最大数目 2)

```
[oracle@racdb01 ~]$ srvctl add serverpool -g mypool -l 0 -u 2
```

将数据库加入到自定义的 Server Pool 里面 :

B, 修改数据库 racdb 的服务器池

```
[oracle@racdb01 ~]$ srvctl modify database -d racdb -g mypool
```

C, 检查 RAC 数据库 racdb 新的 Policy

```
[grid@racdb02 ~]$ srvctl config database -d racdb
Database unique name: racdb
Database name: racdb
Oracle home: /app/product/oracle/11.2.0.4/db_1
```

Oracle user: oracle
Spfile: /app/product/oracle/11.2.0/db_1/dbs/spfileracdb1.ora
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools: mypool
Database instances:
Disk Groups: ASM_DATA,ASM_FRA
Mount point paths:
Services: rac_first,rac_second
Type: RAC
Database is policy managed
发现数据库已经使用 Policy Managed 选项了。

特别提醒：

如果没有特别强烈需要就不用自定义加 Policy Managed，Admin Managed 能兼容 11g RAC 和之前的版本，更通用。

D，使用 crsctl 检查服务器池的状态

已变更为

```
[grid@racdb02 ~]$ crsctl status serverpool -p  
NAME=Free  
IMPORTANCE=0  
MIN_SIZE=0  
MAX_SIZE=-1  
SERVER_NAMES=  
PARENT_POOLS=  
EXCLUSIVE_POOLS=  
ACL=owner:grid:rw,pgroup:oinstall:rw,other::r-x
```

NAME=Generic
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:r-x,pgrp:oinstall:r-x,other::r-x

NAME=ora.mypool
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=2
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r—

E，使用 srvctl 检查服务器池的状态

```
[grid@racdb01 ~]$ srvctl config serverpool -g Free  
Server pool name: Free  
Importance: 0, Min: 0, Max: -1  
Candidate server names:
```

```
[grid@racdb01 ~]$ srvctl config serverpool -g Generic  
PRKO-3160 : Server pool Generic is internally managed as part of administrator-  
managed database configuration and therefore cannot be queried directly via  
srvpool object.
```

```
[grid@racdb01 ~]$ srvctl config serverpool -g mypool  
Server pool name: mypool
```

Importance: 0, Min: 0, Max: 2

Candidate server names:

注意:MIN_SIZE 属性指定资源的基数(数据库等), 假设 min_size 为 2, 数据库实例可以运行在服务器池的两台服务器上。

另一个重要的注意事项: 使用 crsctl 添加服务器池到集群(警告: 使用 crsctl 添加服务器池将对应用服务器等非数据库资源效力, 对数据库资源, 需使用 srvctl 创建服务器池, 请参考文档 [here](#))

F, 添加服务器池

For non-database resources,

```
[grid@racdb02 ~]$ crsctl add serverpool sp1 -attr "MIN_SIZE=1, MAX_SIZE=1, IMPORTANCE=1" -f
```

CRS-2673: Attempting to stop 'ora.racdb.db' on 'racdb01'

CRS-2677: Stop of 'ora.racdb.db' on 'racdb01' succeeded

检查服务器池状态

```
[grid@racdb02 ~]$ crsctl status serverpool -p
```

NAME=Free

IMPORTANCE=0

MIN_SIZE=0

MAX_SIZE=-1

SERVER_NAMES=

PARENT_POOLS=

EXCLUSIVE_POOLS=

ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r-x

NAME=Generic

IMPORTANCE=0

MIN_SIZE=0

MAX_SIZE=-1

SERVER_NAMES=

PARENT_POOLS=

EXCLUSIVE_POOLS=

ACL=owner:grid:r-x,pgrp:oinstall:r-x,other::r-x

NAME=ora.mypool
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=2
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--

NAME=sp1
IMPORTANCE=1
MIN_SIZE=1
MAX_SIZE=1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r—

删除服务器池：

```
[grid@racdb02 ~]$ crsctl delete serverpool sp1
```

```
[grid@racdb02 ~]$ crsctl add serverpool sp2 -attr "MIN_SIZE=1, MAX_SIZE=1,  
IMPORTANCE=2"
```

删除服务器池：

注意：观察差异，当使用 crsctl 时，你不能指定个别的或你希望的主机，而当使用 srvctl 时可以。

G, 查看服务器状态

```
[grid@racdb01 ~]$ crsctl status server -f
```

NAME=racdb01

STATE=ONLINE

ACTIVE_POOLS=Generic ora.racdb ora.racdb_rac_first

STATE_DETAILS=

NAME=racdb02

STATE=VISIBLE

ACTIVE_POOLS=Generic ora.racdb ora.racdb_rac_second

STATE_DETAILS=

转换 Policy-Managed Database 为 Administrator-Managed Database

A, 检查所有服务和数据库的当前配置(如果犯错需要恢复, 那么你可以知道当你开始时配置如何), 如下:

```
[oracle@racdb01 ~]$ srvctl config database -d racdb
```

Database unique name: racdb

Database name: racdb

Oracle home: /app/product/oracle/11.2.0.4/db_1

Oracle user: oracle

Spfile: /app/product/oracle/11.2.0/db_1/dbs/spfileracdb1.ora

Domain:

Start options: open

Stop options: immediate

Database role: PRIMARY

Management policy: AUTOMATIC

Server pools: mypool

Database instances:

Disk Groups: ASM_DATA,ASM_FRA

Mount point paths:

Services: rac_first,rac_second

Type: RAC

Database is policy managed

文档说:

你不能直接转换 a policy-managed database 为 administrator-managed database。而是，你可以使用 `srvctl remove database` 和 `srvctl remove service` 命令删除 policy-managed 配置，然后使用 `srvctl add database` 和 `srvctl add instance` 命令注册该数据库为一个 administrator-managed database。一旦你注册了数据库和实例，必需使用 `srvctl add service` 命令添加回服务。

1.使用 SRVCTL 工具删除数据库

```
[oracle@racdb01 ~]$ srvctl remove database -d racdb
```

```
PRKO-3141 : Database racdb could not be removed because it was running
```

如果数据库正在运行可以使用-f(force)删除正在运行的数据库。但是不推荐使用该方式。

停止数据库并删除

```
[oracle@racdb01 ~]$ srvctl remove database -d racdb
```

```
PRKO-3141 : Database racdb could not be removed because it was running
```

```
[oracle@racdb01 ~]$ srvctl stop database -d racdb
```

```
[oracle@racdb01 ~]$ srvctl status database -d racdb
```

```
Instance racdb_1 is not running on node racdb01
```

```
Instance racdb2 is not running on node racdb02
```

```
[oracle@racdb01 ~]$ srvctl remove database -d racdb
```

```
Remove the database racdb? (y/[n]) y
```

```
[oracle@racdb01 ~]$ srvctl status database -d racdb
```

```
PRCD-1120 : The resource for database racdb could not be found.
```

```
PRCR-1001 : Resource ora.racdb.db does not exist
```

2.添加 administrator-managed 数据库

```
[oracle@racdb01 ~]$ srvctl add database -d racdb -o
```

```
/app/product/oracle/11.2.0.4/db_1 -y automatic
```

```
[oracle@racdb01 ~]$ srvctl config database -d racdb
```

```
Database unique name: racdb
```

```
Database name:
```

```
Oracle home: /app/product/oracle/11.2.0.4/db_1
```

```
Oracle user: oracle
```

```
Spfile:
```

```
Domain:
```

```
Start options: open
```

```
Stop options: immediate
```

```
Database role: PRIMARY
```

```
Management policy: AUTOMATIC
```

Server pools: racdb
Database instances:
Disk Groups:
Mount point paths:
Services:
Type: RAC
Database is administrator managed

3.添加数据库实例

```
[oracle@racdb01 ~]$ srvctl add instance -d racdb -i racdb1 -n racdb01
[oracle@racdb01 ~]$ srvctl add instance -d racdb -i racdb2 -n racdb02
[oracle@racdb01 ~]$ srvctl start database -d racdb
[oracle@racdb01 ~]$ srvctl status database -d racdb
Instance racdb1 is running on node racdb01
Instance racdb2 is running on node racdb02
[oracle@racdb01 ~]$ srvctl config database -d racdb
Database unique name: racdb
Database name:
Oracle home: /app/product/oracle/11.2.0.4/db_1
Oracle user: oracle
Spfile:
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools: racdb
Database instances: racdb1,racdb2
Disk Groups: ASM_DATA,ASM_FRA
Mount point paths:
Services:
Type: RAC
Database is administrator managed
```

如果有非默认 service 需使用 `srvctl add service` 命令添加回服务

4.你必需在最后一步配置 Oracle Enterprise Manager

Services for Policy- and Administrator-Managed Databases

You can define services for both policy-managed and administrator-managed databases.

Services for a policy-managed database are defined to a server pool where the database is running.

Services for policy-managed databases can be defined as:

UNIFORM (running on all instances in the server pool)

SINGLETON (running on only one instance in the pool)

For singleton services, RAC chooses on which instance in the server pool the service is active.

Services for an administrator-managed database define which instances normally support that service.

These are known as the **PREFERRED** instances.

Instances defined to support a service if the preferred instance fails are known as **AVAILABLE** instances.

Oracle Services

To manage workloads or a group of applications, you can define services that you assign to a particular application or to a subset of an application's operations. You can also group work by type under services. For example, online users can use one service, while batch processing can use another, and reporting can use yet another service to connect to the database.

It is recommended that all users who share a service have the same service-level requirements. You can define specific characteristics for services and each service can be a separate unit of work. There are many options that you can take advantage of when using services. Although you do not have to implement these options, using them helps optimize application performance. You can define services for both policy-managed and administrator-managed databases.

Do not use DBMS_SERVICE with cluster-managed services. When Oracle Clusterware starts a service, it updates the database with the attributes stored in the CRS resource. If you use DBMS_SERVICE to modify the service and do not update the CRS resource, the next time CRS resource is started, it will override the database attributes set by DBMS_SERVICE.

Services for Policy- and Administrator-Managed Databases

It is recommended that all users who share a service have the same service-level requirements. You can define specific characteristics for services and each service can be a separate unit of work. There are many options that you can take advantage of when using services. Although you do not have to implement these options, they help optimize application performance. You can define services for both policy-managed and administrator-managed databases.

- **Policy-managed database:** When you define services for a policy-managed database, you define the service to a server pool where the database is running. You can define the service as either uniform (running on all instances in the server pool) or singleton (running on only one instance in the server pool). For singleton services, RAC chooses on which instance in the server pool the service is active. If that instance fails, then the service fails over to another instance in the pool. A service can only run in one server pool.

- **Administrator-managed database:** When you define a service for an administrator-managed database, you define which instances support that service. These are known as the PREFERRED instances. You can also define other instances to support a service if the service's preferred instance fails. These are known as AVAILABLE instances.

Create Service with Enterprise

Administration-Managed

Policy-Managed

Create Services with Enterprise Manager

From your Cluster Database home page, click the Availability tab, and then click Cluster Managed Database Services. On the Cluster Managed Database Services page, click Create Service.

Use the Create Service page to configure a new service in which you do the following:

Select the desired service policy for each instance configured for the cluster database.

Select the desired service properties. Refer to the section “Service Attributes” in this lesson for more information about the properties you can specify on this page.

If your database is administration managed, the High Availability Configuration section allows you to configure preferred and available servers. If your database employs policy-managed administration, you can configure the service cardinality to be UNIFORM or SINGLETON and assign the service to a server pool.

You can also define the management policy for a service. You can choose either an automatic or a manual management policy.

Automatic: The service always starts when the database starts.

Manual: Requires that the service be started manually. Prior to Oracle RAC 11g Release 2, all services worked as though they were defined with a manual management policy.

Note: Enterprise Manager now generates the corresponding entries in your tnsnames.ora files for your services. Just click the “Update local naming parameter (tnsnames.ora) file” check box when creating the service.

Create Services with SRVCTL

To create a service called GL with preferred instance RAC02 and an available instance RAC01:

```
$ srvctl add service -d PROD1 -s GL -r RAC02 -a RAC01
```

To create a service called AP with preferred instance RAC01 and an available instance RAC02:

```
$ srvctl add service -d PROD1 -s AP -r RAC01 -a RAC02
```

To create a SINGLETON service called BATCH using server pool SP1 and a UNIFORM service called ERP using server pool SP2 :

```
$ srvctl add service -d PROD2 -s BATCH -g SP1 \  
-c singleton -y manual  
  
$ srvctl add service -d PROD2 -s ERP -g SP2 \  
-c UNIFORM -y manual
```

Create Services with SRVCTL

For the example in the slide, assume a two-node, administration-managed database called PROD1 with an instance named RAC01 on one node and an instance called RAC02 on the other. Two services are created, AP and GL, to be managed by Oracle Clusterware. The AP service is defined with a preferred instance of RAC01 and an available instance of RAC02.

If RAC01 dies, the AP service member on RAC01 is restored automatically on RAC02. A similar scenario holds true for the GL service.

Note that it is possible to assign more than one instance with both the -r and -a options. However, -r is mandatory but -a is optional.

Next, assume a policy-managed cluster database called PROD2. Two services are created, a SINGLETON service called BATCH and a UNIFORM service called ERP. SINGLETON services run on one of the active servers and UNIFORM services run on all active servers of the server pool. The characteristics of the server pool determines how resources are allocated to the service.

Note: When services are created with srvctl, tnsnames.ora is not updated and the service is not started.

Manage Services with srvctl

Start a named service on all configured instances:

```
$ srvctl start service -d orcl -s AP
```

Disable a service at a named instance:

```
$ srvctl stop service -d orcl -s AP
```

Set an available instance as a preferred instance:

```
$ srvctl disable service -d orcl -s AP -i orcl4
```

```
$ srvctl modify service -d orcl -s AP -i orcl5
```

```
r
```

```
$ srvctl relocate service -d orcl -s AP -i orcl5  
t orcl4
```

Manage Services with srvctl

The slide demonstrates some management tasks with services by using SRVCTL.

Assume that an AP service has been created with four preferred instances: orcl1, orcl2, orcl3, and orcl4. An available instance, orcl5, has also been defined for AP.

In the first example, the AP service is started on all instances. If any of the preferred or

available instances that support AP are not running but are enabled, then they are started.

The stop command stops the AP service on instances orcl3 and orcl4. The instances themselves are not shut down, but remain running possibly supporting other services. The AP service continues to run on orcl1 and orcl2. The intention might have been to perform maintenance on orcl4, and so the AP service was disabled on that instance to prevent automatic restart of the service on that instance. The OCR records the fact that AP is disabled for orcl4. Thus, Oracle Clusterware will not run AP on orcl4 until the service is enabled.

The next command in the slide changes orcl5 from being an available instance to a preferred one. This is beneficial if the intent is to always have four instances run the service because orcl4 was previously disabled. The last example relocates the AP service from instance orcl5 to orcl4. Do not perform other service operations while the online service modification is in progress.

Note: For more information, refer to the *Oracle Real Application Clusters Administrator's Guide*.

Use Services with Client Applications

```
ERP=(DESCRIPTION=          ## Using the SCAN ##
  (LOAD_BALANCE=on)
    (ADDRESS=(PROTOCOL=TCP) (HOST=cluster01-scan) (PORT=1521))
    (CONNECT_DATA=(SERVICE_NAME=ERP)) )
```

```
ERP=(DESCRIPTION=          ## Using VIPs ##
  (LOAD_BALANCE=on)
    (ADDRESS=(PROTOCOL=TCP) (HOST=node-1vip) (PORT=1521))
    (ADDRESS=(PROTOCOL=TCP) (HOST=node-2vip) (PORT=1521))
    (ADDRESS=(PROTOCOL=TCP) (HOST=node-3vip) (PORT=1521))
    (CONNECT_DATA=(SERVICE_NAME=ERP)) )
```

```
url="jdbc:oracle:oci:@ERP"          ## Thick JDBC ##
```

```
url="jdbc:oracle:thin:@(DESCRIPTION=          ## Thin JDBC ##
  (LOAD_BALANCE=on)
    (ADDRESS=(PROTOCOL=TCP) (HOST=cluster01-scan) (PORT=1521))
    (CONNECT_DATA=(SERVICE_NAME=ERP)) ) "
```

Use Services with Client Applications

The first example in the slide shows the TNS connect descriptor that can be used to access the ERP service. It uses the cluster's Single Client Access Name (SCAN). The SCAN provides a single name to the clients connecting to Oracle RAC that does not change throughout the life of the cluster, even if you add or remove nodes from the cluster. Clients connecting with SCAN can use a simple connection string, such as a thin JDBC URL or EZConnect, and still achieve the load balancing and client connection failover. The second example uses virtual IP addresses as in previous versions of the Oracle Database.

The third example shows the thick JDBC connection description using the previously defined TNS connect descriptor.

The third example shows the thin JDBC connection description using the same TNS connect descriptor as the first example.

Note: The LOAD_BALANCE=ON clause is used by Oracle Net to randomize its progress through the protocol addresses of the connect descriptor. This feature is called client connection load balancing.

Services and Connection Load Balancing

The two load balancing methods that you can implement are:

Client-side load balancing: Balances the connection requests across the listeners

Server-side load balancing: The listener directs a connection request to the best instance currently providing the service by using the load balancing advisory (LBA).

FAN, Fast Connection Failover, and LBA depend on a connection load balancing configuration that includes setting the connection load balancing goal for the service.

The load balancing goal for the service can be either:

LONG: For applications having long-lived connections. This is typical for connection pools and SQL*Forms sessions.

SHORT: For applications that have short-lived connections

```
srvctl modify service -s service_name -j LONG|SHORT
```

Services and Connection Load Balancing

Oracle Net Services provides the ability to balance client connections across the instances in an Oracle RAC configuration. You can implement two types of load balancing: client-side and server-side. Client-side load balancing balances the connection requests across the listeners. With server-side load balancing, the listener directs a connection request to the best instance currently providing the service by using the load balancing advisory. In a RAC database, client connections should use both types of connection load balancing.

FAN, Fast Connection Failover, and the load balancing advisory depend on an accurate connection load balancing configuration that includes setting the connection load balancing goal for the service. You can use a goal of either LONG or SHORT for connection load balancing. These goals have the following characteristics:

LONG: Use the LONG load balancing method for applications that have long-lived connections. This is typical for connection pools and SQL*Forms sessions. LONG is the default connection load balancing goal. The following is an example of modifying a service, POSTMAN, with the srvctl utility to define the connection load balancing goal for long-lived sessions:

```
srvctl modify service -s POSTMAN -j LONG
```

SHORT: Use the SHORT connection load balancing method for applications that have short-lived connections. The following example modifies the ORDER service , using `srvctl` to set the goal to SHORT:

```
srvctl modify service -s ORDER -j SHORT
```

Services and Transparent Application Failover

Services simplify the deployment of Transparent Application Failover (TAF).

You can define a TAF policy for a service and all connections using this service will automatically have TAF enabled.

The TAF setting on a service can be **NONE**, **BASIC**, or **PRECONNECT** and overrides any TAF setting in the client connection definition.

To define a TAF policy for a service, the `srvctl` utility can be used as shown below:

Where `-z` is the number of retries, `-w` is the delay between retry attempts and `-j` is the connection load balancing goal.

```
srvctl modify service -s gl.example.com -q TRUE -P BASIC  
-e SELECT -z 180 -w 5 -j LONG
```

Services and Transparent Application Failover

When Oracle Net Services establishes a connection to an instance, the connection remains open until the client closes the connection, the instance is shut down, or a failure occurs. If you configure TAF for the connection, then Oracle Database moves the session to a surviving instance when an outage occurs.

TAF can restart a query after failover has completed but for other types of transactions, such as INSERT, UPDATE, or DELETE, the application must roll back the failed transaction and resubmit the transaction. You must re-execute any session customizations, in other words, ALTER SESSION statements, after failover has occurred. However, with TAF, a connection is not moved during normal processing, even if the workload changes over time.

Services simplify the deployment of TAF. You can define a TAF policy for a service, and all connections using this service will automatically have TAF enabled. This does not require any client-side changes. The TAF setting on a service overrides any TAF setting in the client connection definition. To define a TAF policy for a service, use the `srvctl` utility as in the following example:

```
srvctl modify service -s gl.example.com -q TRUE -P BASIC -e SELECT -z 180 -w 5 -j
```

LONG

Note: TAF applies only to an admin-managed database and not to policy-managed databases.