Oracle 10G 安装配置规范

V1.0

# 前言

## 编写目的

为加强数据库平台建设工作的规范化管理，梳理了数据库平台基础设施设计的相关文档，并进行了深化、细化，力求结合实际的设计、实施工作，对设计、实施起到规范、指导作用。

本规范主要从一个设计者的角度进行阐述，相关章节也按此思路编写。作为一个设计者，首先要了解产品可实现的部署模式，如何选择部署模式，其次要考虑设计涉及到的因素，有针对性地做好数据库的安装配置等。

在界线的划分上，基础产品只涉及本产品的设计，上层应用产品对基础产品的需求放在应用产品中。

在编写过程中，特别关注可操作性，不仅仅是要求，而是提出建议，尽量覆盖设计工作中涉及的工作要点。本规范中参数建议值是对系统设计时的指导，是合理的经验值，但由于应用系统的复杂性，每个系统有自己的特点，建议按建议值进行系统的初始配置，在压力测试和系统上线后根据实际需要做相应的调整。

## 前提和假设

如非特别说明，本规范叙述的产品特性、操作命令、用户界面均基于Oracle 10G，可能有不适用于其它版本的情况。如遇此情况，请参考其它版本厂商技术手册中对应的说明。

## 预期读者

项目基础设施可行性研究、设计和实施人员，应用系统设计人员，相关运行维护技术人员。

# 数据库规划部署

## 数据库类型特点分析

在创建和规划一个Oracle数据库之前，首要任务应确定将来投产的数据库属于何种业务类型。目前的应用业务有以下类型：

* OLTP(Online Transaction Processing)
* OLAP（Online Analytiacl Processing）
* BATCH
* DSS（Decision Support System）
* Hybrid

### OLTP（联机事务处理）

OLTP系统是一个包含繁重及频繁执行的DML应用，其面向事务的活动主要包括更新，同时也包括一些插入和删除。经典的例子是预定系统或在线实时交易系统，例如网上银行和ATM自动取款机系统。OLTP系统可以允许有很高的并发性（在这种情况下，高并发性通常表示许多用户可以同时使用一个数据库系统）。

### OLAP（联机分析处理）

OLAP系统可提供分析服务。这意味着数学、统计学、集合以及大量的计算，一个OLAP系统并不永远适合OLTP或DSS模型，有时它是两者之间的交叉。另外，也可以把OLAP看作是在OLTP系统或DSS之上的一个扩展或一个附加的功能层次。例如，地理信息系统或有关空间的数据库和OLAP数据库相集成，提供图表的映射能力。用于社会统计的人口统计数据库也是一个很好的例子。

### BATCH（批处理系统）

批作业处理系统是作用于数据库的非交互性的自动应用。它通常含有繁忙DML语句并有较低的并发性（在这种情况下，较低的并发性通常表示少数几个用户能够同时使用一个数据库系统），该业务系统会在某一时段，大批量数据（少则几万，多则几十万，几百万条数据）更新/插入/删除该数据库。事务查询的比率决定了如何物理地设计它，经典的例子是与DW有关的成品数据库和可操作数据库，如：操作型数据存储系统（ODS）。

### DSS（决策支持系统）

DSS系统通常是一个大型的、包含历史性内容的只读数据库，通常用于简单的固定查询或特别查询。DSS常常按某种方式变成一个VLDB（Very Large Database）或DW（Data Warehouse）。VLDB的例子如：企业资源管理财务系统（ERP）数据库，该数据库是一个长期存储数据库的历史数据库；DW的例子如：整个集团的工资和人事数据库。

### Hybrid（混合类型系统）

同时数据库系统的应用类型可能是OLTP、OLAP、BATCH等的混合体。也意味着同时拥有上述业务类型特征，这就要求数据库管理员、应用系统分析员、操作系统管理员整体统筹考虑各种业务性能需求及功能需求，对这个系统制定出满足各种业务类型需求的规划，如：企业客户信息整合（ECIF）系统。

## 数据库规模

对于数据库的规模，仅从数据量来衡量其规模的大小。因为数据量的规模是反映数据库规模的主要指标。具体如下：

* 数据库业务数据量小于100GB 属小规模数据库；
* 数据库业务数据量100GB-600GB属中等规模数据库；
* 数据库业务数据量600GB-1TB 属大规模数据库；
* 数据库业务数据量大于1TB 属超大规模数据库。

## 数据库高可用模式

在Oracle 10g 第 1 版，Oracle 的集群件产品可用于所有操作系统，成为了 Oracle RAC 所需的集群技术。随着 Oracle 数据库 10g 第 2 版 (10.2) 的发布，Cluster Ready Services 被重命名为 Oracle Clusterware。如果使用 Oracle 10g 或更高版本，Oracle Clusterware 是大多数运行 Oracle RAC 的平台所需的唯一集群件（不包括 Tru 集群，在这种情况下，您需要供应商集群件）。我们仍然可以使用其他供应商的集群件（如果该集群件通过了认证），但是请记住，**Oracle RAC 仍然需要 Oracle Clusterware**，因为它与数据库软件完全集成。

由于Oracle RAC已经是成熟产品，安装时都应选择RAC模式，除非有特殊情况，可以采用HA模式。

## 安装必要条件

参见文档《169706.1》

## 网络规划原则

### 物理网卡需求

管理网络：1个网卡

业务网络：每个服务器2个网卡。需要为业务网络配置2\*节点数量。**业务网络建议采用BONDING技术**。每个业务网络适配器**必须支持 TCP/IP**。

心跳网络：每个服务器2个网卡。**心跳网络建议经过交换机互联**，不建议使用交叉线互联。**对于心跳网络应采用BONDING技术，互连必须支持用户数据报协议 (UDP)，**使用支持 TCP/IP的高速网络适配器和交换机。UDP被用来RAC之间的通信，TCP被用来CLUSTER之间的通信。**心跳网络建议使用10G网络，最小配置为1G网络。**

每个网络中网络适配器的相关业务接口名称在所有节点上都应相同，网络适配器的相关心跳接口名称在所有节点上都应相同。

**所有的IP地址都应该是静态配置，不应该是DHCP方式获取。**

**如果使用网卡绑定技术，不要使用模式3和 6，模式3会造成通信性能的下降，模式6会造成虚拟IP漂移时出现网络问题。**

**所用的网卡名称不可以包含“.”;**

**在10g及其之前，我们必须使用IPv4协议。**

### 主机名命名规则

RAC机器需要在/etc/hosts中配置各节点IP的别名，命名规则如下**（主机名中包含”\_”是不被建议和支持的）**：

Public IP：如无特殊需求，建议使用dbxx的格式，其中xx为主机序列号；

Private IP: 如无特殊需求，建议使用dbxx-pri的格式，其中xx为主机序列号。

Virtual IP：如无特殊需求，建议使用dbxx-vip的格式，其中xx为主机序列号。

### 监听端口选择

**网络监听建议不要使用默认端口，该规范建议使用同一端口1815.**

### 数据库连接模式

#### 专用服务器连接

服务器只要有足够的资源（CPU和RAM）就使用专用服务器连接。

#### 共享服务器连接

共享服务器只适用于OLTP系统，并且只在资源（CPU和RAM）不足的情况下使用；其他高级连接特性需要使用共享服务器连接时，可以采用共享服务器连接模式。

## 磁盘布局原则

### 存储类型选择

建议选择SAN存储网络，作为共享存储。

由于ORACLE数据库需要确认对数据文件的写操作已经完全写入的磁盘，**因此不建议使用NFS文件系统**，除非NFS存储厂商已经通过ORACLE认证（如REDHAT，NETAPP）。如果使用NFS,rsize和wsize至少为16384，ORACLE 的建议指为32768.

**集群设备不支持directnfs。**

### 磁盘IO调度算法

对于数据库系统建议使用Deadline调度算法。

Deadline调度算法对一个请求的多方面特性进行权衡来进行调度，以期即能满足块设备扇区的顺寻访问又兼顾到一个请求不会在队列中等待太久导致饿死。Deadline调度算法为了兼顾这两个方面，引入了四个队列，这四个队列可分为两类，每一类都由读和写两种队列组成。一类队列用来对请求按起始扇区序号进行排序，通过红黑树来组织，称为sort\_list；另一类对请求按它们的生成时间进行排序，由链表来组织，称为fifo\_list。

如果数据库安装在虚拟化环境下，建议使用noop调度算法，该算法利用少量的CPU资源进行基础排序和整合操作。在虚拟化环境下，分为主机系统和客户机系统，应该使磁盘调度算法尽量在主机层面完成，因此我们建议在虚拟客户机内使用noop调度算法。

### RAID级别规划

在数据库服务器磁盘空间充分冗余的条件下，Raid (1+0)+HotSpare作为优先第一配置选项；如果磁盘空间不是很充足，Raid 5作为第二配置选项。

说明：

★越多表示在本组（Disk\*N）范围内推荐级别越高

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Raid\磁盘** | **Disk\*2** | **Disk\*3** | **Disk\*4** | **Disk\*5** | **Disk\*6** | **Disk\*7** | **Disk\*8** | **Disk\*9** |
| Raid 1 | ★ |  |  |  |  |  |  |  |
| Raid 1+HotSpare |  | ★★ |  |  |  |  |  |  |
| Raid 5 |  | ★ | ★ | ★ | ★ | ★ | ★ | ★ |
| Raid 5+HotSpare |  |  | ★★ | ★★ | ★★ | ★★ | ★★ | ★★ |
| Raid (1+0) |  |  | ★★★ |  | ★★★ |  | ★★★ |  |
| Raid (1+0)+HotSpare |  |  |  | ★★★ |  | ★★★ |  | ★★★ |
|  |  |  |  |  |  |  |  |  |

Raid磁盘组往往会在多个物理磁盘之间对数据进行条带化处理，例如，一般情况下条带大小为64K。当磁盘分区和条带化大小可以对齐时，可以改善磁盘性能，尤其是在开启了DIRECTIO特性数据库系统下。

基于磁盘容量和条带化对齐的考虑，我们建议使用parted工具进行磁盘分区。因为fdisk工具默认情况下会分配前63个sector用于保存磁盘元数据分分区表数据，紧随其后是分区数据，显然此时分区不是从64k开始的，因为每个sector的大小默认为512字节。通过使用parted可以手工指定分区开始位置，因此我们建议使用如下方式进行分区

/sbin/parted /dev/mapper/asm1 mklabel msdos unit s mkpart primary "128 -1"

### 本地磁盘规划

为保证软件安装过程顺利进行，需要保证本地磁盘具有足够的存储空间。具体要求如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **本地磁盘空间需求** | | | |
| **序号** | **用途** | **目录** | **大小** |
| 1 | CRS\_HOME | /u01/app/ oracle/product/10.2.0/crs\_1 | /u01 不小于10G，建议>50G  建议为/u01创建单独分区，不要使用LVM |
| 2 | ORACLE\_HOME | /u01/app/ oracle/product/10.2.0/db\_1 |
| 3 | SOFTWARE\_DIR | /u01/software |
| 4 | 临时目录 | /tmp | 最小1G,建议10G |

### 共享磁盘规划

Oracle 10g中引入了ASM技术，相对于裸设备ASM带来了不少的管理便利性。如果安装最新版本的ORACLE10GR2，建议可以使用ASM,如果是较早版本，可能会存在bug较多的情况。本文以ORACLE10GR2最新版本为基础，因此使用ASM作为存储文件系统。

虽然Oracle 10GR2支持使用O\_DIRECT访问块设备，但是在安装过程中使用块设备OUI会报错，需要使用RAW设备进行安装，安装完成后可以通过命令替换为块设备，详细参加文档《401132.1》。**因此，我们建议：在10G中，表决盘等磁盘仍然使用裸设备，数据文件等使用ASM**。使用ASM时，建议不要使用过多的磁盘组，通常为数据文件创建磁盘组DATA1，为闪回恢复期创建磁盘组FRA1。**建议闪回区大小为数据区大小的两倍**

**不管采用何种技术，强烈建议对使用到的磁盘先分区在使用。**

**裸设备磁盘空间规划**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **存储组件** | **裸设备** | **建议大小** | **最小大小** | **存储内容** |
| VOTE1 | /dev/raw/raw1 | 300M | 280M | VOTEDISK |
| VOTE2 | /dev/raw/raw2 | 300M | 280M | VOTEDISK |
| VOTE3 | /dev/raw/raw3 | 300M | 280M | VOTEDISK |
| OCR1 | /dev/raw/raw4 | 300M | 280M | OCR |
| COR2 | /dev/raw/raw5 | 300M | 280M | OCR |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ASM磁盘空间规划** | | | | | |
| **存储组件** | **ASM卷组** | **卷大小** | **冗余级别** | **AU SIZE** | **存储内容** |
| 数据库文件 | +DATA1 | 500G | EXTERNAL |  | DATAFILE LOG CONTROL OCR |
| 快速恢复区 | +FRA1 | 500G | EXTERNAL |  | ARCHIVELOG CONTROL OCR |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **磁盘命名规划** | | | | | |
| **存储组件** | **ASM卷组** | **ASM\_磁盘名** | **DM多路径** | **UDEV设备** | **ASMLIB设备** |
| VOTE1 |  |  | RACVOTE1 | ORADISK/RACVOTE1 | RACVOTE1 |
| VOTE2 |  |  | RACVOTE2 | ORADISK/RACVOTE2 | RACVOTE2 |
| VOTE3 |  |  | RACVOTE3 | ORADISK/RACVOTE3 | RACVOTE3 |
| OCR1 |  |  | RACOCR1 | ORADISK/RACOCR1 | RACOCR1 |
| COR2 |  |  | RACOCR2 | ORADISK/RACOCR1 | RACOCR2 |
| 数据库文件 | +DATA1 | DATA1P1 | RACDATA1 | ORADISK/RACDATA1 | RACDATA1 |
| 快速恢复区 | +FRA1 | DATA2P1 | RACDATA2 | ORADISK/RACDATA2 | RACDATA2 |

## 内存使用原则

### ASM内存分配原则

由于ASM实例业务简单，内存参数的设置值大于其所需内存空间的和，提供一定的备用内存，**通常建议内存总大小为256M**。

该参数的设置需要考虑冗余性配置和可用磁盘空间。

### DB内存分配原则

ORACLE数据库专用主机中内存使用包括：

* SGA，共享内存全局区：数据库一启动就ORACLE就申请占用，可以通过ipcs -a查看其大小。查看SGA\_MAX\_SIZE参数可以得到其设置值。这部分内存在数据库实例运行期间不会改变；
* PGA，进程全局区：查看PGA\_AGGREGATE\_TARGET可以得到其设置值。这部分内存并非一启动就分配，而是随着应用的使用变化而变化，有排序、并行、HASH JOIN时会增大，甚至会突破设置值，而长时间不使用时又会返回给操作系统；
* 连接到数据库的每个TCP连接本身会占用8M～10M左右内存，对HP Itantium，还会占用14M左右的交换空间。对于使用中间件共享连接池的应用，连接数基本固定，内存占用也就固定；对于直接连接数据库的应用，这部分内存占用随着数据库session数的增减而增减；
* 操作系统自身内存的占用，包括常用软件的内存占用。这部分占用无法给出固定值，简单计算可以按30%物理内存计算，且不低于2.4G，不高于15G。例如对于HP Itantium机器，安装Oracle数据库后操作系统内存占用会达到4G左右。
* RAC的CLuster软件内存占用基本固定，其中代码段大约为0.75G。

因此为新上线数据库预估分配内存时，需要为TCP连接、操作系统预留出足够的内存。可研时，可以按连接数\*10\*1.5/1000估算主机内存需求。

新系统性能测试时初始计划内存分配办法：总物理内存扣除连接数占用的内存及操作系统、常用软件占用的内存之后，剩余的内存按3:1的比例分配给数据库的SGA、PGA, 也即SGA占其中的3/4，PGA占1/4。SGA中SHARED POOL分1/4(且不大于4G，不小于1G)，DB BUFFER和其他内存池共用其余的1/2。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **估算主机内存及初始计划内存分配可参考表** | | | | | | | |
| **最大连接数(个)** | | **最大连接占用内存（G）** | **操作系统等占用（G）** | **Cluster内存占用（G）** | **ASM内存占用（G）** | **主机物理内存配置（G）** | **PGA设置（G）** | **SGA\_MAX\_SIZE设置（G）** |
| <=400 | | 6 | 4.8 | 0.75 | 0.25 | 32 | 5.05 | 15.15 |
| <=1000 | | 15 | 10 | 0.75 | 0.25 | 64 | 9.5 | 28.5 |
| <=3000 | | 45 | 15 | 0.75 | 0.25 | 128 | 16.75 | 50.25 |
| <=8000 | | 120 | 15 | 0.75 | 0.25 | 256 | 30 | 90 |
| 100 | | 1.5 | 2.4 | 0.75 | 0.25 | 8 | 0.775 | 2.325 |



注意：

**RAC系统中需要考虑节点故障时，单节点接过所有负载连接的情况，因此每节点内存需求量都按最大全部连接数考虑。**

此处连接数占用内存的估计都是按照上限值进行的估算，因此比较保守，也预留了日后扩展的需求。实际可连接数比上述数值要大。

### 交换空间分配原则

根据ORACLE官方文档**(Doc ID 225451.1**)，建议配置linux下SWAP空间遵循如下原则。

|  |  |  |  |
| --- | --- | --- | --- |
| **RHEL5** | | | |
| **Amount of RAM in the System** | **Recommended Amount of Swap Space** | | |
| 4GB of RAM or less | a minimum of 2GB of swap space | | |
| 4GB to 16GB of RAM | a minimum of 4GB of swap space | | |
| 16GB to 64GB of RAM | a minimum of 8GB of swap space | | |
| 64GB to 256GB of RAM | a minimum of 16GB of swap space | | |
| 256GB to 512GB of RAM | a minimum of 32GB of swap space | | |
| **RHEL6** | | | |
| **Amount of RAM in the System** | | **Recommended swap space** | **Recommended swap space if allowing for hibernation** |
| ⩽ 2GB | | 2 times the amount of RAM | 3 times the amount of RAM |
| > 2GB – 8GB | | Equal to the amount of RAM | 2 times the amount of RAM |
| > 8GB – 64GB | | 0.5 times the amount of RAM | 1.5 times the amount of RAM |
| > 64GB | | 4GB of swap space | No extra space needed |

### Hugepage使用原则

在 Linux 环境中实施 HugePage 能够极大地提高内核性能。对于内存较大的系统，效果尤其明显。一般而言，**所有 RAM 大于 12GB或者SGA空间超过8G的系统都适合使用 Hugepage**。系统中的 RAM 越大，系统启用 Hugepage 后获得的好处也越大。这是因为内核为映射和维护内存页表所要做的工作量会随着系统内存的增大而增加。启用 Hugepage 能够显著地降低内核要管理的页面数，而且能提高系统的效率。经验表明，如果未启用 Hugepage，内核挤占关键的 Oracle Clusterware 或 Real Application Clusters 守护进程的情况会很常见，而这会导致实例或节点驱逐出现。

**对于ASM实例可以不使用Hugepage技术。**

### /dev/shm设置原则

在ORACLE 11G之前，我们一般不需要设置/dev/shm。

### Linux信号量设置

可以将Semaphores理解为计数器，该计数器用于进程或者线程之间对共享资源的访问同步。根据ORACLE文档《184821.1》，对ORACLE数据库系统的信号量设置，遵循如下计算方法：

|  |  |  |
| --- | --- | --- |
| **信号量设置** | | |
| SEMMNI | 128 | Maximum number of semaphore sets in the entire system. |
| SEMMSL | 250 | Minimum recommended value. SEMMSL should be 10 plus the largest PROCESSES parameter of any Oracle database on the system. |
| SEMMNS | 32000 | Maximum semaphores on the system. This setting is a minimum recommended value. SEMMNS should be set to the sum of the PROCESSES parameter for each Oracle database, add the largest one twice, plus add an additional 10 for each database. This value shoud be equal (SEMMSL\*SEMMNI) |
| SEMOPM | 100 | Maximum number of operations per semop call.SEMOPM should be equal SEMMSL |

## 数据库选项规划

随着Oracle数据库版本的不断升级，Oracle的功能也越来越多，因此安全检查点也越来越多。因此最好根据需求只安装所需内容。如：不需要web 功能，就不要安装相应组件。

在安装数据库软件时，定制安装并仅选择如下组件：

**Oracle Database**

**Enterprise Edition options**

**Oracle partitioning**

**Oracle Net Services**

**Oracle Net Listeners**

**Oracle Call Interface(OCI)**

对如下组件，除非开发中心有特殊需求，否则不安装：

Advanced Security

OLAP

Spatial

Oracle Enterprise Manager Console DB

Programmer

XML Development

利用dbca创建数据库时，在advance component选项页中，点击解除对其中所有选项的选中，包括JVM、SPATIAL、XDB，除非开发中心有特殊需求

## 数据库物理结构规划

### 数据库实例的命名规则

普通使用模式的Oracle数据库的服务名和实例名（SID）是相同的；RAC模式下的Oracle数据库的服务名与实例名不同。

数据库服务名的命名格式为：XXXXYY{m}，不得超过8个字符

数据库的SID的命名格式为：XXXXYY{m}{n}

说明：

其中XXXX表示长度为4个字符的应用项目缩写；

* YY：代表数据库用途，pd代表生产库，hi代表历史库，rp代表报表库，cf代表配置库，st代表Standby灾备库，其他可自行定义，不得超过2个字符；
* m表示数据库序号，从0-9，根据项目的数据库数量进行编号；
* n表示RAC节点实例序号1，2，3……。用以区分多节点的RAC数据库的不同实例。对于普通模式的数据库，该位不指定。

### 表空间设计

生产数据库不得有如TOOLS、XDB、EXAMPLE等oracle默认安装表空间。本文推荐使用ASM，不得使用文件系统和裸设备。

**数据文件使用固定大小，不得设置为自动扩展**,自动扩展语句会造成数据文件的自动增长，在使用ASM的情况下可能造成ASM系统无空闲空间。

数据库管理员必须使用本地管理中的自动分配(Autoallocate)的方式创建表空间，Segment的管理使用ASSM方式。默认Storage设置Minextents为1M。

OLTP系统：每个数据文件最大不超过32G（Oracle理论上限：BLOCK\_SIZE为8K时，数据文件最大32G，16K为64G，32K为128G），超过32G的表空间扩容需要新增数据文件，不足32G的表空间扩容可以直接resize增大数据文件大小。

OLAP系统：由于数据量大，空间需求多，单个数据文件的大小允许设置为128G。**如果单个表空间预期大小超过2T，可以直接使用BigFile表空间**。此时该表空间只能有一个文件，需要增加表空间大小时，直接resize文件大小即可，该文件的最大大小可达到32T（8k Blocks）、64T（16k Blocks）、128T（32k Blocks）。

|  |  |  |
| --- | --- | --- |
| **表空间命名规范** | | |
| **表空间种类** | **表空间命名规则（建议）** | **数据文件命名规则(裸设备，后缀为raw)** |
| TABLES公用表空间 | D\_<功能模块名称>\_nn | D\_<功能模块名称>\_nn\_mmm.dbf |
| TABLES PARTITION分区表空间 | D\_<数据表名>\_nn | D\_<数据表名>\_nn\_mmm.dbf |
| INDEXS公用索引表空间 | I\_<功能模块名称>\_nn | I\_<功能模块名称>\_nn\_mmm.dbf |
| INDEXS PARTITION大表索引空间 | I\_<数据表名>\_nn | I\_<数据表名>\_nn\_mmm.dbf |
| LOB表空间 | B\_<功能模块名称>\_nn | B\_<功能模块名称>\_nn\_mmm.dbf |
| TEMP表空间 | TEMP\_<功能模块名称> | TEMP\_<功能模块名称>\_mmm.dbf |

说明：

nn为从01开始计数的两位整数序号，表示表空间的数目。如：01，02，03，04。。。。。。；

mmm为从001开始计数的三位整数序号，表示数据文件的数目。如：001，002，003，004。。。。。。；

**Undo/temp表空间的估算**

Undo设置原则：oracle9i以后的版本，推荐使用UNDO TABLESPACE，让系统自动管理回滚段。须考虑以下几个问题：

系统并发事务数有多少？

系统是否存在大查询或者大是事务？频繁与否？

能提供给系统的回滚段表空间的磁盘空间是多少？

UNDO表空间初始设置不小于8G，SGA\_MAX\_SIZE大于8G的，至少为SGA\_MAX\_SIZE\*1.2。

Temp设置原则：可创建缺省临时表空间TEMP，取数据库的缺省参数。业务用户默认使用的TEMP表空间初始设置为总空间的5％，且不小于4G，通常不大于100G。

### 控制文件设计

每个数据库实例应至少有两个控制文件，且每个文件存储在独立的物理磁盘上。如果有一个磁盘失效而导致控制文件不可用，与其相关的数据库实例必须关闭。一旦失效的磁盘得到修复，可以把保存在另一磁盘上的控制文件复制到该盘上。这样数据库实例可重新启动。并通过非介质恢复操作使数据库得到恢复。

为了使整个系统的高可靠地运行，**要求至少设置2个控制文件**。

### 日志文件设计

重做日志文件的尺寸会对数据库的性能产生重要影响，因为它的尺寸大小决定着数据库的写进程（DBWn）和日志归档进程（ARCn）。一般情况下，较大的日志文件提供较好的数据库性能，较小的重做日志文件会增加核查点（checkpoint）的活动，从而导致性能的降低。当然为了防止I/O争用，还应把各个重做日志文件分布到不同的物理磁盘上。

不可能为重做日志文件提供特定大小的建议，重做日志文件在几百兆字节到几GB字节都被认为是合理的。欲确定数据库重做日志文件的大小，**应根据该系统产生重做日志的数量，并依据每二十分钟发生一次日志切换这个大致原则来决定，业务高峰时段，日志切换频率不高于五分钟一次。**在系统运行后，我们从alert文件获取日志的切换时间，并根据切换的间隔来调整重组日志的大小。

日志文件命名规范：

格式：redo\_tt\_gg\_mm.dbf (如果为裸设备，后缀为.raw)

tt为从01开始计数的两位整数序号，表示thread#。如：01，02，03，04。。。。。。；

gg为从01开始计数的两位整数序号，表示group#。如：01，02，03，04。。。。。。；

mm为从01开始计数的两位整数序号，表示组内redo的数目。如：01，02，03，04。。。。。。；

### 归档日志

Oracle可以将联机日志文件保存到多个不同的位置，将联机日志转换为归档日志的过程称之为归档，相应的日志被称为归档日志。归档日志（archivelog ）文件存储方法：

放在ASM单独的磁盘组+FRA1，独立于+DATA1磁盘组。对RAC系统来说，各自实例使用相同的ASM磁盘组，方便RMAN进行归档日志备份。

**要求归档日志的ARCH磁盘组大小可以保证容纳2天产生的归档日志。**

## 字符集规划

|  |  |  |
| --- | --- | --- |
| **NO** | **PARAMETER** | **VALUE** |
| 1 | NLS\_LANGUAGE | AMERICAN |
| 2 | NLS\_NCHAR\_CHARACTERSET | AL16UTF16 |
| 3 | NLS\_TERRITORY | AMERICA |
| 4 | NLS\_CURRENCY | $ |
| 5 | NLS\_ISO\_CURRENCY | AMERICA |
| 6 | NLS\_NUMERIC\_CHARACTERS | ., |
| 7 | NLS\_CHARACTERSET | ZHS16GBK |
| 8 | NLS\_CALENDAR | GREGORIAN |
| 9 | NLS\_DATE\_FORMAT | DD-MON-RR |
| 10 | NLS\_DATE\_LANGUAGE | AMERICAN |
| 11 | NLS\_SORT | BINARY |
| 12 | NLS\_TIME\_FORMAT | HH.MI.SSXFF AM |
| 13 | NLS\_TIMESTAMP\_FORMAT | DD-MON-RR HH.MI.SSXFF AM |
| 14 | NLS\_TIME\_TZ\_FORMAT | HH.MI.SSXFF AM TZR |
| 15 | NLS\_TIMESTAMP\_TZ\_FORMAT | DD-MON-RR HH.MI.SSXFF AM TZR |
| 16 | NLS\_DUAL\_CURRENCY | $ |
| 17 | NLS\_COMP | BINARY |
| 18 | NLS\_LENGTH\_SEMANTICS | BYTE |
| 19 | NLS\_NCHAR\_CONV\_EXCP | FALSE |

# 安装先决条件准备

## 检查操作系统兼容性

|  |
| --- |
| **ORACLE 10g操作系统认证列表** |
|  |

如果使用EL4系列的操作系统，建议升级到4.7之后，以避免如下bug：6116137、6125546、5041764、5136660等。

## 检查集群软件兼容性

|  |
| --- |
| **ORACLE 10g RAC 认证列表** |
|  |

## 用户及组

执行以下过程，验证在安装节点上存在用户 nobody,nobody用户会用于执行oracle的某些任务：

[root@localhost ~]# id nobody

uid=99(nobody) gid=99(nobody) 组=99(nobody)

如果该命令显示了 nobody 用户的信息。如果用户 nobody 不存在，则输入以下命令进行创建：

[root@localhost ~]# useradd -M -u 99 -g nobody -s /sbin/nologin nobody

创建Oracle用户，命令如下：

groupadd -g 1000 oinstall

groupadd -g 1300 dba

groupadd -g 1301 oper

useradd -m -u 1100 -g oinstall -G dba,oper -d /home/oracle -s /bin/bash oracle

设置 Oracle 帐户的口令：

echo oracle | passwd --stdin oracle

## 配置HugePage

配置/etc/security/limits.conf,设置memlock值(KB),该值可略小于物理内存。重启系统后配置生效。

cat > /tmp/limits.sh <<EOF

#!/bin/bash

memtotal\_kb=`grep 'MemTotal' /proc/meminfo | awk '{print \$2 }'`

#the total amount of memory pages that can be used system wide,in pages

memlock=\$(( \$memtotal\_kb \* 4 / 5 ))

softml=\`grep -v "^#" /etc/security/limits.conf | grep "memlock" | grep "soft" | wc -c\`

hardml=\`grep -v "^#" /etc/security/limits.conf | grep "memlock" | grep "hard" | wc -c\`

if [ \$softml != 0 ]

then

sed -i "s/.\*soft.\*memlock.\*/\\* soft memlock \$memlock/" /etc/security/limits.conf

else

echo "\* soft memlock \$memlock" >> /etc/security/limits.conf

fi

if [ \$hardml != 0 ]

then

sed -i "s/.\*hard.\*memlock.\*/\\* hard memlock \$memlock/" /etc/security/limits.conf

else

echo "\* hard memlock \$memlock" >> /etc/security/limits.conf

fi

EOF

使用oracle用户通过如下命令查看，配置是否生效。

ulimit –l

配置/etc/sysctl.conf,修改vm.nr\_hugepages参数值为SGA/hugepagesize+4。hugepagesize在linux下默认为2M。

在kernel2.4下，需要使用参数vm.hugetlb\_pool。

查看transparent\_hugepage是否启用，如果目录不存在或者返回结果为never，则表示transparent\_hugepage没有启用，否则我们需要禁用transparent\_hugepage。

cat /sys/kernel/mm/transparent\_hugepage/enabled

修改/etc/grub.conf在kenel后添加如下选项transparent\_hugepage=never，以禁用transparent\_hugepage。

对于RHEL7如果上述方法行不通，可以在/usr/lib/tuned/throughput-performance/tuned.conf文件中添加如下选项来禁用transparent\_hugepage.

[vm]

transparent\_hugepage=never

## 配置临时目录/TMP

Oracle的安装要求/tmp目录至少有400MB以上的空余空间，**安装RAC时，临时目录不可以使用共享目录**。可以用以下语句检查/tmp目录的空间：

# df /tmp

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/hda2 113836004 40094512 67958896 38% /

如果没发现/tmp目录没有足够的空间了，可以在另外创建一个目录，并修改ORACLE的环境变量，方法如下：

su - root

mkdir /u01/tmp

chown root.root /u01/tmp

chmod 777 /u01/tmp

export TEMP=/u01/tmp       # used by Oracle

export TMPDIR=/u01/tmp     # used by Linux programs like the linker "ld"

当完成了Oracle安装后，可以先关闭Oracle服务器，然后移除刚才创建的目录，并复位相关环境变量。

su - root

rmdir /u01/tmp

unset TEMP

unset TMPDIR

## 配置SWAP

根据交换空间分配原则，检查当前系统的SWAP空间分配是否合理。

  检查系统内存：

# grep MemTotal /proc/meminfo

MemTotal: 511940 kB (1034688 kB=1G)

检查系统交换区：

# cat /proc/swaps

Filename Type Size Used Priority

/dev/hda3 partition 1044216 256584 -1

  如果发现交换空间太小，可以有两种方式增加交换区，一种是采用文件方式，一种是采用新建分区的方式。

使用文件增加交换区：

# cat /proc/swaps

Filename Type Size Used Priority

/dev/hda3 partition 1044216 256584 -1

# dd if=/dev/zero of=tmpswap bs=1k count=524288

524288+0 records in

524288+0 records out

# chmod 600 tmpswap

# mkswap tmpswap

Setting up swapspace version 1, size = 536866 kB

# swapon tmpswap

# cat /proc/swaps

Filename Type Size Used Priority

/dev/hda3 partition 1044216 263628 -1

/tmpswap file 524280 0 -2

# 删除交换区：

# swapoff tmpswap

# rm -f tmpswap

# cat /proc/swaps

Filename Type Size Used Priority

/dev/hda3 partition 1044216 263624 -1

## 配置内核参数

cat > /tmp/sysctl.sh <<EOF

#!/bin/bash

function setSysCtl(){

echo "set sysctl variable: " \$1,"to: " \$2

ccnt=\`grep -v "^#" /etc/sysctl.conf | grep "\$1" | wc -c\`

if [ \$ccnt != 0 ]

then

sed -i "s/[^#]\*\$1.\*/\$1=\$2/" /etc/sysctl.conf

else

echo "\$1=\$2" >> /etc/sysctl.conf

fi

}

pagesize=\`getconf PAGE\_SIZE\`

memtotal\_kb=\`grep 'MemTotal' /proc/meminfo | awk '{print \$2 }'\`

memtotal=\$(( 1024 \* \$memtotal\_kb ))

shmmni=4096

#the total amount of memory pages that can be used system wide,in pages

shmall=\$(( \$memtotal \* 3 / 5 / \$pagesize ))

if [ \$shmall -lt 2097152 ]

then

shmall=2097152

fi

#the maximum shared memory segment allowed by the kernel,in bytes

shmmax=\$(( \$shmall \* \$pagesize ))

setSysCtl "kernel.shmmax" "\$shmmax"

setSysCtl "kernel.shmall" "\$shmall"

setSysCtl "kernel.shmmni" "\$shmmni"

echo "Please input the processes count："

read process

SEMMNI=142

SEMMSL=\$(( \$process + 10 ))

if [ \$SEMMSL -lt 250 ]

then

SEMMSL=250

fi

SEMMNS=\$(( \$SEMMNI \* \$SEMMSL ))

SEMOPM=\$SEMMSL

setSysCtl "kernel.sem" "\$SEMMSL \$SEMMNS \$SEMOPM \$SEMMNI"

setSysCtl "fs.aio-max-nr" "1048576"

setSysCtl "fs.file-max" "6815744"

setSysCtl "net.core.wmem\_max" "1048576"

setSysCtl "net.core.wmem\_default" "262144"

setSysCtl "net.core.rmem\_max" "2097152"

setSysCtl "net.core.rmem\_default" "262144"

setSysCtl "net.ipv4.tcp\_rmem" "4096 262144 4194304"

setSysCtl "net.ipv4.tcp\_wmem" "4096 262144 262144"

setSysCtl "net.ipv4.tcp\_keepalive\_time" "30"

setSysCtl "net.ipv4.tcp\_keepalive\_intvl" "60"

setSysCtl "net.ipv4.tcp\_keepalive\_probes" "9"

setSysCtl "net.ipv4.tcp\_retries2" "3"

setSysCtl "net.ipv4.tcp\_syn\_retries" "2"

setSysCtl "net.ipv4.ip\_local\_port\_range" "9000 65500"

setSysCtl "net.ipv4.conf.default.rp\_filter" "2"

setSysCtl "kernel.randomize\_va\_space" "0"

setSysCtl "kernel.exec-shield" "0"

setSysCtl "vm.min\_free\_kbytes" "524288"

setSysCtl "fs.suid\_dumpable" "1"

setSysCtl "kernel.core\_uses\_pid" "1"

setSysCtl "net.ipv4.icmp\_echo\_ignore\_broadcasts" "0"

setSysCtl "vm.swappiness" "0"

setSysCtl "vm.dirty\_background\_ratio" "3"

setSysCtl "vm.dirty\_ratio" "15"

setSysCtl "vm.dirty\_expire\_centisecs" "500"

setSysCtl "vm.dirty\_writeback\_centisecs" "100"

setSysCtl "vm.hugetlb\_shm\_group" "1000"

EOF

## 配置Core File Creation

修改/etc/profile文件添加如下内容

cat >> /etc/profile <<EOF

if [ \$USER = "oracle" ]; then

ulimit -S -c unlimited > /dev/null 2>&1

fi

EOF

## 配置PAM

修改/etc/pam.d/login 文件中添加或编辑下面一行内容（如果不存在此行）：

cat >> /etc/pam.d/login <<EOF

session required     pam\_limits.so

EOF

如果是64为系统，则

cat >> /etc/pam.d/login <<EOF

session required /lib64/security/pam\_limits.so

EOF

## 修改SELINUX

sed -i 's#=enforcing#=disabled#g' /etc/selinux/config

## 安装软件包

将操作系统的安装镜像文件上传到服务器的/mnt目录下，修改镜像文件名称为linux.iso。在/mnt目录下创建文件夹linux,并挂载linux.iso至linux目录

mv \*.iso linux.iso

mkdir /mnt/linux

mount -t iso9660 /mnt/linux.iso /mnt/linux -o loop

修改yum配置文件，进入如下目录/etc/yum.repos.d，在该目录创建文件夹bak，将所有的.repo文件移动到bak文件夹下。

cd /etc/yum.repos.d

mkdir bak

mv \*.repo ./bak/

创建local.repo文件。

cd /etc/yum.repos.d

touch local.repo

cat >> /etc/yum.repos.d/local.repo <<EOF

[local]

name=local

baseurl=file:///mnt/linux

enable=1

gpgcheck=0

EOF

使用如下命令安装所需的软件包,**glibc必须要2.5-24版本或以上**

yum install autoconf -y

yum install binutils.x86\_64 -y

yum install compat-db.x86\_64 -y

yum install compat-gcc-34-c++ -y

yum install compat-gcc-34.x86\_64 -y

yum install compat-libstdc++-296.i386 -y

yum install compat-libstdc++-33.i386 -y

yum install compat-libstdc++-33.x86\_64 -y

yum install elfutils-libelf-devel.x86\_64 -y

yum install elfutils-libelf.x86\_64 -y

yum install gcc-c++.x86\_64 -y

yum install gcc.x86\_64 -y

yum install gdb -y

yum install gdbm -y

yum install glibc-common.x86\_64 -y

yum install glibc-devel.i386 -y

yum install glibc-devel.x86\_64 -y

yum install glibc-headers.x86\_64 -y

yum install glibc.x86\_64 -y

yum install glibc.i686 -y

yum install libaio-devel.x86\_64 -y

yum install libaio.x86\_64 -y

yum install libaio.i386 -y

yum install libgcc.x86\_64 -y

yum install libgcc.i386 -y

yum install libgomp.x86\_64 -y

yum install libstdc++.x86\_64 -y

yum install libstdc++-devel.x86\_64 -y

yum install libXp.x86\_64 -y

yum install libXp.i386 -y

yum install libXtst.i386 -y

yum install make -y

yum install libgnome.x86\_64 -y

yum install libgnomeui.x86\_64 -y

yum install openmotif -y

yum install openssl.i686 -y

yum install openssh-clients -y

yum install openssh -y

yum install pdksh -y

yum install screen -y

yum install setarch -y

yum install sysstat.x86\_64 -y

yum install unixODBC -y

yum install unixODBC-devel -y

yum install unixODBC.i386 -y

yum install unixODBC-devel.i386 -y

yum install unzip -y

yum install util-linux.x86\_64 -y

yum install vim-enhanced -y

yum install xdpyinfo -y

yum install xorg-x11-server-Xorg -y

yum install xorg-x11-xinit -y

以下为图形界面，视情况进行安装

yum install vnc -y

yum install vnc-server -y

yum groupinstall "X Window System" -y

yum groupinstall "GNOME Desktop Environment" -y

## 关闭相关服务

service iptables stop

service ip6tables stop

service NetworkManager stop

service atd stop

service lvm2-monitor stop

service cups stop

service mdmonitor stop

service autofs stop

service cpuspeed stop

service kdump stop

service nfs stop

service nfslock stop

service netfs stop

service bluetooth stop

service iscsi stop

service iscsid stop

service libvirt-guests stop

service libvirtd stop

service rpcbind stop

service acpid stop

chkconfig anacron off

chkconfig apmd off

chkconfig atd off

chkconfig autofs off

chkconfig bluetooth off

chkconfig cpuspeed off

chkconfig cups-config-daemon off

chkconfig cups off

chkconfig gpm off

chkconfig haldaemon off

chkconfig iiim off

chkconfig ip6tables off

chkconfig iptables off

chkconfig iscsi off

chkconfig iscsid off

chkconfig isdn off

chkconfig kdump off

chkconfig libvirt-guests off

chkconfig libvirtd off

chkconfig lvm2-monitor off

chkconfig mdmonitor off

chkconfig netfs off

chkconfig NetworkManager off

chkconfig nfs off

chkconfig nfslock off

chkconfig pcmcia off

chkconfig portmap off

chkconfig readahead off

chkconfig readahead\_early off

chkconfig rhnsd off

chkconfig rpcbind off

chkconfig rpcgssd off

chkconfig rpcidmapd off

chkconfig smartd off

chkconfig avahi-daemon off

## 修改运行级别

sed -i 's#id:5:initdefault#id:3:initdefault#g' /etc/inittab

## HangCheck模块

Oracle9202以前，都是使用watchdog来检查系统是否挂起。从9202以后，oracle使用hangcheck-timer模块。hangcheck-timer模块被加载到内核中，能够定时检查系统是否挂起。

使用如下命令检查该模块是否存在：

find /lib/modules -name "hangcheck-timer.ko"

使用相应的参数加载该模块：

/sbin/insmod hangcheck-timer hangcheck\_tick=1 hangcheck\_margin=10 hangcheck\_reboot=1

两个参数的含义：

hangcheck\_tick：该参数定义了模块启动的间隔时段。其默认值为60秒。Oracle建议将其设置为1秒。

hangcheck\_margin：该参数定义了多少时间以后系统没响应就认为是挂起。以秒为单位。其默认值为180秒。Oracle建议将其设置为10秒。

hangcheck\_reboot：该参数定义hangcheck模块是否会重启操作系统，建议设置为1，即可以重启操作系统。

检查该模块是否被成功加载：

grep Hangcheck /var/log/messages

为了在linux重启以后，自动加载该模块，可以将加载模块的命令添加到/etc/modules.conf中

cat >> /etc/rc.local <<EOF

/sbin/modprobe hangcheck-timer hangcheck\_tick=1 hangcheck\_margin=10 hangcheck\_reboot=1

EOF

## 配置用户shell limits

cat >> /etc/security/limits.conf <<EOF

oracle soft nproc 16384

oracle hard nproc 16384

oracle soft nofile 65536

oracle hard nofile 65536

EOF

对于 Bourne、Bash 或 Korn shell，将以下行添加到 /etc/profile 文件：

cat >> /etc/profile <<EOF

if [ \$USER = "oracle" ]; then

    if [ \$SHELL = "/bin/ksh" ]; then

        ulimit -p 16384

        ulimit -n 65536

    else

        ulimit -u 16384 -n 65536

    fi

    umask 022

fi

EOF

对于 C shell（csh 或 tcsh），将以下行添加到 /etc/csh.login 文件：

if($USER=="oracle") then

limit maxproc 16384

limit descriptors 65536

endif

## 配置Oracle Profile

cat > /home/oracle/.bash\_profile <<EOF

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

        . ~/.bashrc

fi

alias ls="ls -FA"

ORACLE\_SID=ORCL1; export ORACLE\_SID

ORACLE\_UNQNAME=ORCL; export ORACLE\_UNQNAME

ORACLE\_BASE=/u01/app/oracle; export ORACLE\_BASE

ORACLE\_HOME=\$ORACLE\_BASE/product/10.2.0/db\_1; export ORACLE\_HOME

# do not use ORA\_CRS\_HOME,some error will happed!!!

CRS\_HOME=\$ORACLE\_BASE/product/10.2.0/crs\_1; export CRS\_HOME

ORACLE\_PATH=/u01/app/common/oracle/sql; export ORACLE\_PATH

ORACLE\_TERM=xterm; export ORACLE\_TERM

NLS\_DATE\_FORMAT="YYYY-MM-DD HH24:MI:SS"; export NLS\_DATE\_FORMAT

TNS\_ADMIN=\$ORACLE\_HOME/network/admin; export TNS\_ADMIN

PATH=\$ORACLE\_HOME/bin:\$CRS\_HOME/bin:\${PATH}

PATH=\${PATH}:/usr/bin:/bin:/usr/bin/X11:/usr/local/bin

PATH=\${PATH}:/u01/app/common/oracle/bin

export PATH

export TEMP=/tmp

export TMPDIR=/tmp

umask 022

# User specific environment and startup programs

EOF

在Oracle RAC 软件的安装过程中，OUI 使用 SSH 运行命令并将文件复制到其他节点。在安装过程中，系统中的隐藏文件（例如 .bashrc 或 .cshrc）如果包含 stty 命令，则会导致生成文件错误和其他安装错误。

要避免该问题，必须在每个 Oracle 安装所有者的用户主目录中修改这些文件以取消所有 STDERR 输出，如下所示：

cat >> /home/oracle/.bash\_profile <<EOF

if [ -t 0 ]; then

 stty intr ^C

fi

EOF

FOR C SHELL

test -t 0  
if ($status == 0) then  
 stty intr ^C  
endif

## 配置JUMBO FRAMES

如果主机和交换机支持jumbo frames，我们建议开启jumbo frames，这样可以改善网络负载并降低cpu利用率。如果需要开启网卡的jumbo frame，修改网卡的配置文件如/etc/sysconfig/network-scripts/ifcfg-eth0,增添如下配置参数

MTU=9000

## 配置ETH1000网卡

在2.6内核下，REDHAT4\REHDAT5系统中，Eth1000网卡默认没有开启FLOW CONTROL。当RAC环境下的网络负载较高时，容易造成INTERCONNECTS丢包。开启FLOW CONTROL的方法如下。

修改/etc/modprobe.conf添加如下内容：

options e1000 FlowControl=1,1

## 配置网络IP

配置IP地址，修改每个节点上的/etc/hosts文件，确保 /etc/hosts 文件的环回地址中不包含节点名。如果 RAC 节点名出现在环回地址中，您在 RAC 安装期间将收到以下错误消息：

ORA-00603: ORACLE server session terminated by fatal error

或者

ORA-29702: error occurred in Cluster Group Service operation

127.0.0.1 localhost.localdomain localhost

10.255.254.121 db01

10.255.254.122 db02

10.255.254.123 db01-vip

10.255.254.124 db02-vip

192.168.1.121 db01-pri

192.168.1.122 db02-pri

修改主机名：

hostname db01

sed -i 's#HOSTNAME=.\*#HOSTNAME=db01#g' /etc/sysconfig/network

echo "NOZEROCONF=YES" >>/etc/sysconfig/network

hostname db02

sed -i 's#HOSTNAME=.\*#HOSTNAME=db03#g' /etc/sysconfig/network

echo "NOZEROCONF=YES" >>/etc/sysconfig/network

## 配置SSH对等性

使用如下语句配置SSH对等性，**注意要逐行拷贝，不要一次性拷贝**。配置完成后要进行测试。

su - oracle

/usr/bin/ssh-keygen -t dsa

cat ~/.ssh/id\_dsa.pub > ~/.ssh/authorized\_keys

ssh oracle@db01 cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

chmod 644 ~/.ssh/authorized\_keys

/usr/bin/ssh-keygen -t dsa

cat ~/.ssh/id\_dsa.pub > ~/.ssh/authorized\_keys

ssh oracle@db02 cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

chmod 644 ~/.ssh/authorized\_keys

**要确保未启用 X11 forwarding**， 如果看到类似这样的消息：“Warning:No xauth data; using fake authentication data for X11 forwarding.”，这意味着您的授权密钥文件配置正确，但是，您的SSH 配置启用了 X11 forwarding。

确保 ForwardX11 属性设为 no。例如，将以下内容插入 ~/.ssh/config 文件：

Host \*

ForwardX11 no

如果我们在配置SSH对等性的过程中输入了密码，则可以通过如下方式配置无密码登陆。

# exec /usr/bin/ssh-agent $SHELL  
# ssh-add  
Enter passphrase for /home/crs/.ssh/id\_rsa  
Identity added: /home/crs/.s

sh/id\_rsa (/home/crs/.ssh/id\_rsa)

## 配置DNS

当需要使用DNS 服务器时，在/etc/resolv.conf添加DNS服务器的信息。

[root@localhost ~]# cat /etc/resolv.conf

nameserver 114.114.114.114

options timeout:1

options attempts:2

## 配置NTP服务

安装ntp软件包，并配置服务自启动。

yum install ntp

chkconfig ntpd on

编辑首节点的ntp.conf文件，使其从外部获取系统时间.

vi /etc/ntp.conf

#New ntp server added by Robinson

server 1.cn.pool.ntp.org prefer # 添加首选的时钟服务器

#只允许192.168.7.\*网段的客户机进行时间同步

restrict 192.168.7.0 mask 255.255.255.255 nomodify notrap

broadcastdelay 0.008

编辑其他节点的ntp.conf文件，使其从首节点获取系统时间.

vi /etc/ntp.conf

#New ntp server added by Robinson

server 10.201.5.201 prefer

broadcastdelay 0.008

编辑节点的/etc/sysconfig/ntpd文件，配置参数如下，在其中设置 -x 标志，这样可避免向后调整时间。

#The following item added by Robinson

#Set to 'yes' to sycn hw clock after successful ntpdate

SYNC\_HWCLOCK=yes #此选项用于自动校准系统时钟与硬件时钟

OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"

节点间必须具有至少一个相同ntp ip，否则PRVF-5416，但是我们可以忽略该错误。

在首节点配置rc.local脚本，使其在启动时使用ntpdate更新时间,通过ntpd获取时间信息，有时会有延时。

## 创建软件目录

Grid的home目录不能在base目录下，oracle的home必须在base目录下，否则会报INS-32018错误。

mkdir -p /u01/backup

mkdir -p /u01/software

mkdir -p /u01/app/oraInventory

mkdir -p /u01/app/oracle

mkdir -p /u01/app/oracle/admin

mkdir -p /u01/app/oracle/product/10.2.0/db\_1

mkdir -p /u01/app/oracle/product/10.2.0/crs\_1

chown -R oracle:oinstall /u01

chown -R oracle:oinstall /u01/app/oracle

chmod -R 755 /u01/

chmod -R 775 /u01/software/

## 虚拟化环下中断设置

默认情况下，在rhel4和rhel5中，时钟中断频率为1000HZ,在虚拟化环境下，该频率可能过高，因此，我们建议将其调低，设置方法如下：

在32位系统下，修改grub.conf,添加如下内核设置

divider=10 clocksource=acpi\_ppm iommu=soft

在64位系统下，修改grub.conf,添加如下内核设置

divider=10 notsc iommu=soft

设置完成后，时间中断频率为1000/10,即100Hz。

## 配置磁盘调度算法

Deadline磁盘调度算法比较适合于ORACLE\RAC MYSQL等数据库类应用。配置Deadline为linux默认磁盘调度算法，需要修改grud.conf文件，添加内核参数如下：

title Red Hat Enterprise Linux Server (2.6.18-8.el5)

root (hd0,0)

kernel /vmlinuz-2.6.18-8.el5 ro root=/dev/sda2 elevator=deadline

initrd /initrd-2.6.18-8.el5.img

在虚拟化环境下，配置noop为linux默认磁盘调度算法，需要修改grud.conf文件，添加内核参数如下：

title Red Hat Enterprise Linux Server (2.6.18-8.el5)

root (hd0,0)

kernel /vmlinuz-2.6.18-8.el5 ro root=/dev/sda2 elevator=noop

initrd /initrd-2.6.18-8.el5.img

## 配置ISCSI存储

在集群中的各个节点上配置 iSCSI 启动器。而创建分区只应在 RAC 集群的一个节点上执行。 Linux 包括 Open-iSCSI iSCSI 软件启动器，该软件启动器位于 iscsi-initiator-utils RPM 中。

### 安装 iSCSI（启动器）服务

查看iscsi-initiator-utils是否安装。

rpm -qa --queryformat "%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n"| grep iscsi-initiator-utils

如果没装安装，则在系统盘中，找到软件包并安装。

rpm -Uvh iscsi-initiator-utils-\*

### 配置 iSCSI（启动器）服务

验证 iscsi-initiator-utils 程序包已经安装到Oracle RAC 节点之后，启动 iscsid 服务，并使其在系统引导时自动启动并自动登录到所需的 iSCSI 目标。

service iscsid start

chkconfig iscsid on

chkconfig iscsi on

使用 iscsiadm 命令行接口发现网络存储服务器上的所有可用目标。这应该在所有Oracle RAC 节点上执行：

[root@danode1 ~]#iscsiadm -m discovery -t sendtargets -p 192.168.1.199

172.16.1.241:3260,0 iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-crs

172.16.1.241:3260,0 iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-data

172.16.1.241:3260,0 iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-fra

手动登录到 iSCSI 目标，这需要在所有Oracle RAC 节点上运行。

[root@danode1 ~]# iscsiadm -m node -T iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-crs -p 172.16.1.241 -l

[root@danode1 ~]# iscsiadm -m node -T iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-data -p 172.16.1.241 -l

[root@danode1 ~]# iscsiadm -m node -T iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-fra -p 172.16.1.241 -l

配置自动登录，在所有 Oracle RAC 节点上执行以下命令：

[root@danode1 ~]# iscsiadm -m node -T iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-crs -p 172.16.1.241 --op update -n node.startup -v automatic

[root@danode1 ~]# iscsiadm -m node -T iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-data -p 172.16.1.241 --op update -n node.startup -v automatic

[root@danode1 ~]# iscsiadm -m node -T iqn.2000-03.com.h3c-.YZBackup-1-SCU1.darac-fra -p 172.16.1.241 --op update -n node.startup -v automatic

如果在ISCSI SERVER上启用Failover的话，注意要在rac节点中修改几个参数。设置超时时间，编辑/etc/iscsi.conf文件，在添加DiscoveryAddress之前添加如下2行：

IdleTimeout=20

ConnFailTimeout=300

DiscoveryAddress=172.16.1.241

## 配置scsi\_id.config

查看/etc/scsi\_id.config文件，确保文件中至少包含如下内容，以便scsi\_id命令按照预期结果执行。

[root@localhost ~]# grep -v ^# /etc/scsi\_id.config

vendor="ATA",options=-p 0x80

options= -g

## 配置MULTIPATH多路径

通过yum功能安装一下软件包：

yum -y install device-mapper device-mapper-multipath

通过命令：lsmod | grep dm\_multipath 检查是否正常安装成功。

使用如下命令加载驱动模块

modprobe dm-multipath

modprobe dm-round-robin

multipath –v2

通过vi命令创建一个Multipath的配置文件路径是/etc/multipath.conf ,在配置文件中添加multipath正常工作的最简配置如下(**RHEL5下，通过multipath.conf可以直接配置多路径权限**，**需要重启操作系统**)：

devices {

device {

vendor "(LSI|NETAPP)"

product "INF-01-00"

path\_grouping\_policy group\_by\_prio

detect\_prio yes

prio rdac

path\_checker rdac

hardware\_handler "1 rdac"

failback manual

features "2 pg\_init\_retries 50"

no\_path\_retry fail

retain\_attached\_hw\_handler yes

}

}

blacklist {

devnode "^(ram|raw|loop|fd|md|dm-|sr|sr|scd|st)[0-9]\*"

devnode "^hd[a-z]"

devnode "^asm/\*"

devnode "ofsctl"

}

multipaths {

multipath {

wwid 360080e500032a3680000051e58b64b6a

alias RACOCR1

gid 1000

uid 0

mode 644

}

}

## 磁盘分区

/sbin/parted /dev/mapper/RACDATA1 mklabel gtk unit s mkpart primary "128 -1"

## 裸设备配置

在RHEL5U4之前，linux不建议使用裸设备，因此需要我们手工在rc.local中配置裸设备和其权限。在RHEL5U4之后，LINUX可以支持使用RAWSERVICE服务配置裸设备。可以使用如下脚本配置裸设备：

cat >> /etc/sysconfig/rawdevices <<EOF

/dev/raw/raw1 /dev/mapper/RACVOTE1p1

/dev/raw/raw2 /dev/mapper/RACVOTE2p1

/dev/raw/raw3 /dev/mapper/RACVOTE3p1

/dev/raw/raw4 /dev/mapper/RACOCR1p1

/dev/raw/raw5 /dev/mapper/RACOCR2p1

EOF

chkconfig rawdevices on

service rawdevices restart

## 配置UDEV

不同版本的操作系统，UDEV的配置方式会略有差异，本文建议使用RHEL5系列的操作系统，如果使用RHEL4系列操作系统，可以参照ORACLE 9I中关于udev的配置。

建议使用udev+asm来配置oracle rac的存储，首先确定安装了udev软件包：

rpm -qa | grep udev

udev-095-14.32.el5

使用scsi\_id命令查看磁盘属性

[root@db201 ~]# scsi\_id -g -u /dev/sdb

360080e500032a3400000051b58b74174

#### DM设备

如果已经使用了DM多路径软件，并且配置了alias，则不需要再使用udev配置磁盘永久名称,。

可以通过如下UDEV规则设置磁盘所属组和权限。设置完成后重启Multipathd服务，即可。

cat > /etc/udev/rules.d/30-oradmowner.rules <<EOF

SUBSYSTEM!="block", GOTO="quickexit"

KERNEL!="dm-\*", GOTO="quickexit"

PROGRAM=="/sbin/dmsetup info -c --noheadings -o name -m %m -j %M"

RESULT=="\*RACOCR\*", OWNER="root", GROUP="oinstall", MODE="0640"

RESULT=="\*RACVOTE\*", OWNER="oracle", GROUP="oinstall", MODE="0660"

RESULT=="\*RACDATA\*", OWNER="oracle", GROUP="oinstall", MODE="0660"

RESULT=="\*RACFRA\*", OWNER="oracle", GROUP="oinstall", MODE="0660"

LABEL="quickexit"

EOF

#### 裸设备

可以使用如下方式，设置裸设备的权限（**RHEL5不再支持单独的权限配置文件/etc/udev/permissions.d/50-udev.permissions**）。

cat > /etc/udev/rules.d/99-orarawowner.rules <<EOF

KERNEL=="raw1", OWNER="oracle" GROUP="oinstall", MODE="0660"

KERNEL=="raw2", OWNER="oracle" GROUP="oinstall", MODE="0660"

KERNEL=="raw3", OWNER="oracle" GROUP="oinstall", MODE="0660"

KERNEL=="raw4", OWNER="root" GROUP="oinstall", MODE="0640"

KERNEL=="raw5", OWNER="root" GROUP="oinstall", MODE="0640"

EOF

#### 其他设备

在其他情况下（如单路径设备），如果需要udev配置磁盘永久名称，可以参考如下方式**(注意：记住将OCR的权限配置为root:oinstall 640,其他磁盘为oracle:oinstall 660)**：

cat > /etc/udev/rules.d/60-oradisk.rules <<EOF

ACTION=="add", KERNEL=="sd\*1", BUS=="scsi", PROGRAM=="/sbin/scsi\_id -g -s /block/$parent", RESULT=="36589cfc00000044e4c79d3e197d154a4" SYMLINK+="ORADISK/RACVOTE1", OWNER="oracle", GROUP="dba", MODE="0660"

EOF

## 安装并配置ASMLIB(不建议)

如果通过前面的设置已经可以固定磁盘名称和磁盘权限，则可以不需要配置ASMLIB。

安装并配置ASMLib 2.0，根据自己的内核版本选择相应的软件包：

oracleasm-2.6.18-194.el5-2.0.5-1.el5.x86\_64.rpm

oracleasmlib-2.0.4-1.el5.x86\_64.rpm

oracleasm-support-2.1.3-1.el5.x86\_64.rpm

配置ASM：

现在，您已安装了用于 Linux 的 ASMLib 程序包，接下来您需要配置并加载 ASM 内核模块。需要在两个 Oracle RAC 节点上作为 root 用户帐户执行该任务。

注：oracleasm 命令的默认路径为 /usr/sbin。以前版本中使用的 /etc/init.d 路径并未被弃用，但该路径下的 oracleasm 二进制文件现在一般用于内部命令

[root@server01 sbin]# oracleasm configure –i

该脚本完成以下任务：

创建 /etc/sysconfig/oracleasm 配置文件

创建 /dev/oracleasm 挂载点

挂载 ASMLib 驱动程序文件系统

注：ASMLib 驱动程序文件系统并非常规的文件系统。它只用于自动存储管理库与自动存储管理驱动程序的通讯。

输入以下命令以加载 oracleasm 内核模块：

[root@server01 sbin]# oracleasm init

Creating /dev/oracleasm mount point: /dev/oracleasm

Loading module "oracleasm": oracleasm

Mounting ASMlib driver filesystem: /dev/oracleasm

创建 ASM 磁盘只需在 RAC 集群中的一个节点上以 root 用户帐户执行。该操作完成后，应在两个 Oracle RAC 节点上运行 oracleasm listdisks 命令以验证是否创建了所有 ASM 磁盘以及它们是否可用。

oracleasm createdisk CRSV1 /dev/disk/by-id/scsi-3600a0b800050cfa2000008af4d38e5e8-part1

注：这里是通过磁盘阵列的ID进行连接，这是因为这个ID是唯一不变的，如果通过硬盘名称如sdc1、sdd1时，如果系统因增加或减少硬件时，名称会发生变化。

在其他节点上扫描磁盘：

oracleasm scandisks

oracleasm listdisks

如果使用了MULTIPATHD多路径，则应该修改/etc/sysconfig/oracleasm如下(否则安装过程执行root.sh会报错)：

Oracleasm\_scanorder="dm"

Oracleasm\_scanexclude="sd"

## X Windows配置（不建议）

Oracle Universal Installer 是一个 GUI 接口，要求使用 X 服务器。在启用了用户等效性的终端会话中（您将从其中执行 Oracle 安装的节点），需要将环境变量 DISPLAY 设置为有效的 X 窗口显示：

[oracle@racnode1 ~]$ DISPLAY=<Any X-Windows Host>:0

[oracle@racnode1 ~]$ export DISPLAY

**将 DISPLAY 变量设置为有效的 X 窗口显示后，要确保未启用 X11 forwarding**， 如果看到类似这样的消息：“Warning:No xauth data; using fake authentication data for X11 forwarding.”，这意味着您的授权密钥文件配置正确，但是，您的 SSH 配置启用了 X11 forwarding。例如：

使用文本编辑器，编辑或创建 ~/.ssh/config 文件

确保 ForwardX11 属性设为 no。例如，将以下内容插入 ~/.ssh/config 文件：

Host \*

ForwardX11 no

**因为通过silent方式安装，速度较快，且不易出错，因此建议不要使用图形化安装。**

## 修改/etc/redhat-release

cp /etc/redhat-release /etc/redhat-release.bak

echo "Red Hat Enterprise Linux Server release 4" > /etc/redhat-release

## 数据库软件包

|  |  |  |
| --- | --- | --- |
| **需要下载软件列表** | | |
| **序号** | **名称** | **MD5校验码** |
| 1 | 10201\_clusterware\_linux\_x86\_64.cpio.gz | bb6ba01c9f1f50612ba0fee0d245237c |
| 2 | 10201\_database\_linux\_x86\_64.cpio.gz | 8f092c0b21081444fd1938c51664b1f0 |
| 3 | p8202632\_10205\_Linux-x86-64.zip | dda60b6319d2242a90ed4a325dbc2d33 |
| 4 | p6880880\_102000\_Linux-x86-64.zip | ae36d7e1cf4205620ed5626e43af377a |
| 5 | p20299014\_10205\_Linux-x86-64.zip | 7f365b92baaf4e6851062b1164473d33 |
| 6 | cvupack\_Linux\_x86\_64.zip | 8231fe7cefbda60b0b09245fe1df8db3 |
| 7 | p4679769\_10201\_Linux-x86-64.zip | 7e944daf8fae68e1052a47534fbe9919 |

**在进行软件安装之前，通过MD5校验码验证软件文件是否正确**。

在确认软件包正确无误后，使用如下命令解压缩软件包。**建议将软件包下载到/u01/software目录下，并直接解压到该文件夹下**。

mkdir -p /u01/software

cd /u01/software/

zcat 10201\_clusterware\_linux\_x86\_64.cpio.gz | cpio -idmv

zcat 10201\_database\_linux\_x86\_64.cpio.gz | cpio -idmv

unzip p8202632\_10205\_Linux-x86-64.zip

unzip p6880880\_102000\_Linux-x86-64.zip

unzip p20299014\_10205\_Linux-x86-64.zip

unzip cvupack\_Linux\_x86\_64.zip -d ./cvupack

unzip p4679769\_10201\_Linux-x86-64.zip

chmod -R 755 p8202632\_10205\_Linux-x86-64

chmod -R 755 database

chmod -R 755 clusterware

# 软件安装

## Cluster安装检测

### 安装配置CVUQDISK

在Oracle RAC 节点上安装操作系统程序包CVUQDISK。如果没有CVUQDISK，集群验证实用程序就无法发现共享磁盘，当运行（手动运行或在 Oracle Cluster Infrastructure 安装结束时自动运行）集群验证实用程序时，您会收到这样的错误消息：“Package cvuqdisk not installed”。

CVUQDISK RPM 包含在 Oracle CRS 安装介质上的 rpm 目录中。以root用户在所有节点上执行如下命令安装CVUQDISK。

cd /u01/software/clusterware/rpm

CVUQDISK\_GRP=oinstall; export CVUQDISK\_GRP

rpm -ivh cvuqdisk-1.0.1-1.rpm

### 检查先决条件

以ORACLE用户，使用 CVU 验证硬件和操作系统设置：

export LANG=en\_US

cd /u01/software/clusterware

./cluvfy/runcluvfy.sh stage -post hwos -n db01,db02 –verbose

因为BUG4437727（在10.2.0.3中解决），我们建议使用最新版的cvupack，来执行验证工作。因此在进行安装验证之前，我们建议将PUBLIC IP进行修改，以使验证过程可以顺利进行。

export LANG=en\_US

cd /u01/software/cvupack/bin

./cluvfy stage -pre crsinst -n db01,db02 -r 10gR2 –verbose

查看 [CVU 报告](http://www.oracle.com/technology/pub/files/hunter_rac11gr2_iscsi_cvu_report_hwos.txt)。CVU 执行的所有其他检查的结果报告应该为“passed”，之后才能继续进行 Oracle Cluster 的安装。

## Cluster安装

首先配置cluster.rsp文件，文件内容如下，将该文件保存为/tmp/cluster.rsp

cat > /tmp/cluster.rsp <<EOF

RESPONSEFILE\_VERSION=2.2.1.0.0

UNIX\_GROUP\_NAME="oinstall"

FROM\_LOCATION="../stage/products.xml"

ORACLE\_HOME="/u01/app/oracle/product/10.2.0/crs\_1"

ORACLE\_HOME\_NAME="OraClusterware10ghome1"

TOPLEVEL\_COMPONENT={"oracle.crs","10.2.0.1.0"}

SHOW\_SPLASH\_SCREEN=false

SHOW\_WELCOME\_PAGE=false

SHOW\_NODE\_SELECTION\_PAGE=false

SHOW\_SUMMARY\_PAGE=false

SHOW\_INSTALL\_PROGRESS\_PAGE=true

SHOW\_CONFIG\_TOOL\_PAGE=true

SHOW\_XML\_PREREQ\_PAGE=false

SHOW\_ROOTSH\_CONFIRMATION=true

SHOW\_END\_SESSION\_PAGE=true

SHOW\_EXIT\_CONFIRMATION=true

NEXT\_SESSION=false

NEXT\_SESSION\_ON\_FAIL=false

SHOW\_DEINSTALL\_CONFIRMATION=false

SHOW\_DEINSTALL\_PROGRESS=false

RESTART\_SYSTEM=false

RESTART\_REMOTE\_SYSTEM=false

SHOW\_END\_OF\_INSTALL\_MSGS=true

COMPONENT\_LANGUAGES={"en","zh\_CN"}

s\_clustername="crs10g"

sl\_tableList={"db01:db01-pri:db01-vip:N:Y","db02:db02-pri:db02-vip:N:Y"}

ret\_PrivIntrList={"eth0:10.255.254.0:1","eth1:192.168.1.0:2"}

n\_storageTypeOCR=1

s\_ocrpartitionlocation="/dev/raw/raw4"

s\_ocrMirrorLocation="/dev/raw/raw5"

n\_storageTypeVDSK=1

s\_votingdisklocation="/dev/raw/raw1"

s\_OcrVdskMirror1RetVal="/dev/raw/raw2"

s\_VdskMirror2RetVal="/dev/raw/raw3"

EOF

调用如下命令进行安装CLUSTER

export LANG=en\_US

cd /u01/software/clusterware

./runInstaller -silent -responseFile /tmp/cluster.rsp -ignoreSysPrereqs

安装的最后阶段，根据提示应该运行相应脚本，在运行这些脚本前，请按照如下步骤修复相关bug。

如果安装环境是10.2.0.4之前和RHEL5。在所有节点修改VIPCA和SRVCTL命令，

sed -i "s#LD\_ASSUME\_KERNEL#XX\_ASSUME\_KERNEL#g" /u01/app/oracle/product/10.2.0/crs\_1/bin/vipca

sed -i "s#LD\_ASSUME\_KERNEL#XX\_ASSUME\_KERNEL#g" /u01/app/oracle/product/10.2.0/crs\_1/bin/srvctl

因为BUG 4679769,在执行root.sh脚本之前，请打上补丁p4679769\_10201\_Linux-x86-64。

cd /u01/software

unzip /u01/software/p4679769\_10201\_Linux-x86-64.zip

cd /u01/app/oracle/product/10.2.0/crs\_1/bin

cp clsfmt.bin clsfmt.bin.bak.`date +%F%T`

rm -rf ./clsfmt.bin

cp /u01/software/4679769/clsfmt.bin ./clsfmt.bin

chmod 755 ./clsfmt.bin

chown oracle:oinstall ./clsfmt.bin

安装系统提示，在各个节点使用root运行如下脚本：

/u01/app/oracle/product/10.2.0/crs\_1/root.sh

当在最后一个节点执行root.sh脚本时，如果发现如下错误：

Error 0(Native: listNetInterfaces:[3])

[Error 0(Native: listNetInterfaces:[3])]

根据文档 414163.1，首先以ORACLE运行脚本

/u01/app/oracle/product/10.2.0/crs\_1/cfgtoollogs/configToolAllCommands

然后以root用户执行如下内容

cd /u01/app/oracle/product/10.2.0/crs\_1/bin

./vipca -silent -nodelist db01,db02 -nodevips db01/db01-vip,db02/db02-vip

最后再次执行如下脚本

/u01/app/oracle/product/10.2.0/crs\_1/cfgtoollogs/configToolAllCommands

验证集群安装

crs\_stat –t

## Database安装检测

安装前验证：

export LANG=en\_US

cd /u01/software/cvupack/bin

./cluvfy stage -pre dbinst -fixup -n db01,db02 -osdba dba –verbose

## 安装DATABASE SOFTWARE

首先配置dbinstall.rsp文件，文件内容如下，将该文件保存为/u01/software/database/dbsoft.rsp

cat > /u01/software/database/dbsoft.rsp <<EOF

RESPONSEFILE\_VERSION=2.2.1.0.0

UNIX\_GROUP\_NAME=oinstall

FROM\_LOCATION="../stage/products.xml"

ORACLE\_HOME="/u01/app/oracle/product/10.2.0/db\_1"

ORACLE\_HOME\_NAME="OraDbHome1"

TOPLEVEL\_COMPONENT={"oracle.server","10.2.0.1.0"}

SHOW\_SPLASH\_SCREEN=false

SHOW\_WELCOME\_PAGE=false

SHOW\_NODE\_SELECTION\_PAGE=false

SHOW\_SUMMARY\_PAGE=false

SHOW\_CUSTOM\_TREE\_PAGE=false

SHOW\_INSTALL\_PROGRESS\_PAGE=true

SHOW\_CONFIG\_TOOL\_PAGE=true

SHOW\_XML\_PREREQ\_PAGE=false

SHOW\_ROOTSH\_CONFIRMATION=true

SHOW\_END\_SESSION\_PAGE=true

SHOW\_EXIT\_CONFIRMATION=true

NEXT\_SESSION=false

NEXT\_SESSION\_ON\_FAIL=false

SHOW\_DEINSTALL\_CONFIRMATION=false

SHOW\_DEINSTALL\_PROGRESS=false

RESTART\_SYSTEM=false

RESTART\_REMOTE\_SYSTEM=false

SHOW\_END\_OF\_INSTALL\_MSGS=true

CLUSTER\_NODES={"db01","db02"}

COMPONENT\_LANGUAGES={"en","zh\_CN"}

INSTALL\_TYPE="Custom"

oracle.server:DEPENDENCY\_LIST={"oracle.rdbms:10.2.0.1.0","oracle.options:10.2.0.1.0","oracle.network:10.2.0.1.0","oracle.sysman.console.db:10.2.0.1.0","oracle.rdbms.oci:10.2.0.1.0","oracle.precomp:10.2.0.1.0","oracle.xdk:10.2.0.1.0","oracle.sqlplus.isqlplus:10.2.0.1.0","oracle.odbc:10.2.0.1.0"}

oracle.options:DEPENDENCY\_LIST={"oracle.network.aso:10.2.0.1.0","oracle.rdbms.partitioning:10.2.0.1.0","oracle.sdo:10.2.0.1.0","oracle.oraolap:10.2.0.1.0"}

oracle.network:DEPENDENCY\_LIST={"oracle.network.listener:10.2.0.1.0"}

s\_nameForDBAGrp=dba

s\_nameForOPERGrp=oper

varSelect=3

EOF

调用如下命令进行安装DATABASE SOFTWARE

export LANG=en\_US

cd /u01/software/database

./runInstaller -silent -responseFile /u01/software/database/dbsoft.rsp -ignoreSysPrereqs

安装完成后，安装系统提示，在各个节点使用root运行如下脚本：

/u01/app/oracle/product/10.2.0/db\_1/root.sh

如果安装环境是10.2.0.4之前和RHEL5。在所有节点修改SRVCTL命令：

sed -i "s#LD\_ASSUME\_KERNEL#XX\_ASSUME\_KERNEL#g" /u01/app/oracle/product/10.2.0/db\_1/bin/srvctl

# 打补丁

## 解压补丁包

oracle用户分别在第一个节点运行如下内容：

cd /u01/software/

unzip /u01/software/p8202632\_10205\_Linux-x86-64.zip

## PS补丁（CRS）

**注意：下面的操作，仅仅需要在第一个节点执行。**

首先生成响应文件，安装补丁时需要。

cat > /tmp/patchcrs.rsp <<EOF

RESPONSEFILE\_VERSION=2.2.1.0.0

UNIX\_GROUP\_NAME=oinstall

FROM\_LOCATION="../stage/products.xml"

ORACLE\_HOME="/u01/app/oracle/product/10.2.0/crs\_1"

ORACLE\_HOME\_NAME="OraClusterware10ghome1"

TOPLEVEL\_COMPONENT={"oracle.patchset.db","10.2.0.5.0"}

SHOW\_SPLASH\_SCREEN=false

SHOW\_WELCOME\_PAGE=false

SHOW\_COMPONENT\_LOCATIONS\_PAGE=false

SHOW\_CUSTOM\_TREE\_PAGE=false

SHOW\_SUMMARY\_PAGE=true

SHOW\_INSTALL\_PROGRESS\_PAGE=true

SHOW\_REQUIRED\_CONFIG\_TOOL\_PAGE=true

SHOW\_OPTIONAL\_CONFIG\_TOOL\_PAGE=true

SHOW\_CONFIG\_TOOL\_PAGE=true

SHOW\_XML\_PREREQ\_PAGE=false

SHOW\_RELEASE\_NOTES=false

SHOW\_END\_OF\_INSTALL\_MSGS=true

SHOW\_ROOTSH\_CONFIRMATION=true

SHOW\_END\_SESSION\_PAGE=false

SHOW\_EXIT\_CONFIRMATION=true

NEXT\_SESSION=false

NEXT\_SESSION\_ON\_FAIL=false

SHOW\_DEINSTALL\_CONFIRMATION=false

SHOW\_DEINSTALL\_PROGRESS=false

ACCEPT\_LICENSE\_AGREEMENT=true

RESTART\_SYSTEM=false

CLUSTER\_NODES={"db01","db02"}

COMPONENT\_LANGUAGES={"en","zh\_CN"}

SECURITY\_UPDATES\_VIA\_MYORACLESUPPORT=false

DECLINE\_SECURITY\_UPDATES=false

COLLECTOR\_UPGRADE=false

COLLECTOR\_IGNORE\_CONFIGURATION=true

EOF

在所有节点，用root用户关闭集群服务

crsctl stop crs

以ORACLE用户进行补丁升级

cd /u01/software/p8202632\_10205\_Linux-x86-64/Disk1/

export LANG=en\_US

./runInstaller -silent -responseFile /tmp/patchcrs.rsp -ignoreSysPrereqs

在所有节点，用root执行如下脚本

/u01/app/oracle/product/10.2.0/crs\_1/install/root102.sh

升级完成后建议重启操作系统。

## PS补丁（DB）

**注意：下面的操作，仅仅需要在第一个节点执行。**

首先生成响应文件，安装补丁时需要。

cat > /tmp/patchdb.rsp <<EOF

RESPONSEFILE\_VERSION=2.2.1.0.0

UNIX\_GROUP\_NAME=oinstall

FROM\_LOCATION="../stage/products.xml"

ORACLE\_HOME="/u01/app/oracle/product/10.2.0/db\_1"

ORACLE\_HOME\_NAME="OraDbHome1"

TOPLEVEL\_COMPONENT={"oracle.patchset.db","10.2.0.5.0"}

SHOW\_SPLASH\_SCREEN=false

SHOW\_WELCOME\_PAGE=false

SHOW\_COMPONENT\_LOCATIONS\_PAGE=false

SHOW\_CUSTOM\_TREE\_PAGE=false

SHOW\_SUMMARY\_PAGE=true

SHOW\_INSTALL\_PROGRESS\_PAGE=true

SHOW\_REQUIRED\_CONFIG\_TOOL\_PAGE=true

SHOW\_OPTIONAL\_CONFIG\_TOOL\_PAGE=true

SHOW\_CONFIG\_TOOL\_PAGE=true

SHOW\_XML\_PREREQ\_PAGE=false

SHOW\_RELEASE\_NOTES=false

SHOW\_END\_OF\_INSTALL\_MSGS=true

SHOW\_ROOTSH\_CONFIRMATION=true

SHOW\_END\_SESSION\_PAGE=false

SHOW\_EXIT\_CONFIRMATION=true

NEXT\_SESSION=false

NEXT\_SESSION\_ON\_FAIL=false

SHOW\_DEINSTALL\_CONFIRMATION=false

SHOW\_DEINSTALL\_PROGRESS=false

ACCEPT\_LICENSE\_AGREEMENT=true

RESTART\_SYSTEM=false

CLUSTER\_NODES={"db01","db02"}

COMPONENT\_LANGUAGES={"en","zh\_CN"}

SECURITY\_UPDATES\_VIA\_MYORACLESUPPORT=false

DECLINE\_SECURITY\_UPDATES=false

COLLECTOR\_UPGRADE=false

COLLECTOR\_IGNORE\_CONFIGURATION=true

EOF

在所有节点，关闭相关服务

srvctl stop asm -n db01

srvctl stop asm -n db02

srvctl stop database -n db01

srvctl stop database -n db02

srvctl stop listener -n db01

srvctl stop listener -n db02

以ORACLE用户进行补丁升级

cd /u01/software/p8202632\_10205\_Linux-x86-64/Disk1/

export LANG=en\_US

./runInstaller -silent -responseFile /tmp/patchdb.rsp -ignoreSysPrereqs –force

在所有节点，用root执行如下脚本，升级完成后建议重启操作系统。

/u01/app/oracle/product/10.2.0/db\_1/root.sh

## 更新patch工具

以ORACLE用户解压补丁更新工具到指定目录

export LANG=en\_US

cd /u01/software/

unzip -o p6880880\_102000\_Linux-x86-64.zip -d $ORACLE\_HOME

## PSU升级

查询是否存在冲突补丁：

cd $ORACLE\_HOME/OPatch

./opatch prereq CheckConflictAgainstOHWithDetail -phBaseDir /u01/software/20299014

RAC环境下，查询是否可以滚动升级(结果：可以滚动升级)

cd $ORACLE\_HOME/OPatch

./opatch query -all /u01/software/20299014 | grep rolling

关闭节点1的数据库实例和asm实例

srvctl stop instance -d ORCL -i ORCL1

srvctl stop asm -n db01 -o asm1

srvctl stop listener -n db01 -l LISTENER\_DB01

应用补丁

cd /u01/software/20299014

$ORACLE\_HOME/OPatch/opatch apply

根据提示关闭节点2的相关实例

srvctl stop instance -d ORCL -i ORCL2

srvctl stop asm -n db02 -o asm2

srvctl stop listener -n db02 -l LISTENER\_DB02

升级程序会远程对节点2进行补丁升级。升级成功后开启所有节点

srvctl start listener -n db01 -l LISTENER\_DB01

srvctl start asm -n db01 -o asm1

srvctl start instance -d ORCL -i ORCL1

srvctl start listener -n db02 -l LISTENER\_DB02

srvctl start asm -n db02 -o asm2

srvctl start instance -d ORCL -i ORCL2

仅仅在节点1运行如下脚本

cd $ORACLE\_HOME/rdbms/admin

sqlplus / as sysdba

SQL> @catbundle.sql psu apply

SQL> -- Execute the next statement only if this is the first PSU applied for 10.2.0.5 or this is the first PSU applied since 10.2.0.5.3.

SQL> @utlrp.sql

执行如下脚本查看补丁情况

select \* from dba\_registry\_history;

## 修改diagwait

为了避免不必要的节点重启（如系统调度延迟等），ORACLE建议将diagwait配置为13。详细信息可以参见文档《559365.1》。

配置过程如下：注意，需要在所有节点关闭CRS服务，参数修改可以只在一个节点进行。

# <CRS\_HOME>/bin/crsctl stop crs

# <CRS\_HOME>/bin/oprocd stop

# ps -ef |egrep "crsd.bin|ocssd.bin|evmd.bin|oprocd"

# <CRS\_HOME>/bin/crsctl set css diagwait 13 -force

# <CRS\_HOME>/bin/crsctl start crs

# <CRS\_HOME>/bin/crsctl get css diagwait

# DBCA创建数据库

## 添加ASM RAC实例

在创建数据库之前，首选配置ASM实例，作为RAC集群，首先要将ASM实例注册到ocr。

su - oracle

srvctl add asm -n db01 -i +ASM1 -o /u01/app/oracle/product/10.2.0/db\_1

srvctl add asm -n db02 -i +ASM2 -o /u01/app/oracle/product/10.2.0/db\_1

## 配置ASM实例

cat > /tmp/asmca.rsp <<EOF

[GENERAL]

RESPONSEFILE\_VERSION = "10.0.0"

OPERATION\_TYPE = "configureASM"

[CONFIGUREASM]

NODELIST="db01,db02"

DISKSTRING="/dev/mapper/\*"

ASM\_SYS\_PASSWORD="1qaz2WSX"

EOF

调用如下命令

export LANG=en\_US

dbca -silent -responseFile /tmp/asmca.rsp

## 创建磁盘组

CREATE DISKGROUP DATA1 EXTERNAL REDUNDANCY

disk

'/dev/mapper/RACDATA1p1' NAME DATA1P1;

CREATE DISKGROUP FRA1 EXTERNAL REDUNDANCY

disk

'/dev/mapper/RACDATA2p1' NAME DATA2P1;

alter diskgroup data1 mount;

alter diskgroup fra1 mount;

## 创建数据库

dbca是ORACLE提供的图形化数据库创建工具，使用该工具可以方便的完成数据库创建工作，由于脚本中不完全包括dbca工具中的动作，**因此要求必须使用dbca来创建**。

在使用DBCA的过程中，会自动创建ASM实例,但是我们还是建议先配置好ASM实例，再建库。

首选生成数据库定制化模板，模板内如如下：

cat > /u01/app/oracle/product/10.2.0/db\_1/assistants/dbca/templates/INST.dbt <<EOF

<DatabaseTemplate name="INST" description="" version="10.2.0.5.0">

<CommonAttributes>

<option name="ISEARCH" value="false"/>

<option name="OMS" value="false"/>

<option name="JSERVER" value="false"/>

<option name="SPATIAL" value="false"/>

<option name="ODM" value="false">

<tablespace id="SYSAUX"/>

</option>

<option name="IMEDIA" value="false"/>

<option name="XDB\_PROTOCOLS" value="false">

<tablespace id="SYSAUX"/>

</option>

<option name="ORACLE\_TEXT" value="false">

<tablespace id="SYSAUX"/>

</option>

<option name="SAMPLE\_SCHEMA" value="false"/>

<option name="CWMLITE" value="false">

<tablespace id="SYSAUX"/>

</option>

<option name="EM\_REPOSITORY" value="false">

<tablespace id="SYSAUX"/>

</option>

<option name="HTMLDB" value="false"/>

</CommonAttributes>

<Variables/>

<CustomScripts Execute="false"/>

<InitParamAttributes>

<InitParams>

<initParam name="pga\_aggregate\_target" value="256" unit="MB"/>

<initParam name="log\_archive\_dest\_1" value="'LOCATION=+DATA1/'"/>

<initParam name="processes" value="150"/>

<initParam name="db\_recovery\_file\_dest\_size" value="2048" unit="MB"/>

<initParam name="sga\_target" value="512" unit="MB"/>

<initParam name="compatible" value="10.2.0.5.0"/>

<initParam name="db\_create\_file\_dest" value="+DATA1"/>

<initParam name="background\_dump\_dest" value="{ORACLE\_BASE}/admin/{DB\_UNIQUE\_NAME}/bdump"/>

<initParam name="job\_queue\_processes" value="10"/>

<initParam name="db\_name" value=""/>

<initParam name="user\_dump\_dest" value="{ORACLE\_BASE}/admin/{DB\_UNIQUE\_NAME}/udump"/>

<initParam name="dispatchers" value="(PROTOCOL=TCP) (SERVICE={SID}XDB)"/>

<initParam name="audit\_file\_dest" value="{ORACLE\_BASE}/admin/{DB\_UNIQUE\_NAME}/adump"/>

<initParam name="db\_domain" value=""/>

<initParam name="open\_cursors" value="300"/>

<initParam name="db\_block\_size" value="8" unit="KB"/>

<initParam name="db\_recovery\_file\_dest" value="+FRA1"/>

<initParam name="log\_archive\_format" value="%t\_%s\_%r.dbf"/>

<initParam name="core\_dump\_dest" value="{ORACLE\_BASE}/admin/{DB\_UNIQUE\_NAME}/cdump"/>

<initParam name="remote\_login\_passwordfile" value="exclusive"/>

<initParam name="undo\_management" value="AUTO"/>

<initParam name="db\_file\_multiblock\_read\_count" value="16"/>

<initParam name="undo\_tablespace" value="UNDOTBS2"/>

</InitParams>

<MiscParams>

<databaseType>MULTIPURPOSE</databaseType>

<maxUserConn>200</maxUserConn>

<percentageMemTOSGA>40</percentageMemTOSGA>

<customSGA>true</customSGA>

<characterSet>AL32UTF8</characterSet>

<nationalCharacterSet>AL16UTF16</nationalCharacterSet>

<archiveLogMode>true</archiveLogMode>

<initParamFileName>{ORACLE\_BASE}/admin/{DB\_UNIQUE\_NAME}/pfile/init.ora</initParamFileName>

</MiscParams>

<SPfile useSPFile="true">+DATA1/{DB\_UNIQUE\_NAME}/spfile{SID}.ora</SPfile>

</InitParamAttributes>

<StorageAttributes>

<ControlfileAttributes id="Controlfile">

<maxDatafiles>1024</maxDatafiles>

<maxLogfiles>192</maxLogfiles>

<maxLogMembers>3</maxLogMembers>

<maxLogHistory>1</maxLogHistory>

<maxInstances>32</maxInstances>

<image name="&lt;OMF\_CONTROL\_0>" filepath="+DATA1/{DB\_UNIQUE\_NAME}/"/>

<image name="&lt;OMF\_CONTROL\_1>" filepath="+DRA1/{DB\_UNIQUE\_NAME}/"/>

</ControlfileAttributes>

<DatafileAttributes id="&lt;OMF\_SYSAUX\_DATAFILE\_0>">

<tablespace>SYSAUX</tablespace>

<temporary>false</temporary>

<online>true</online>

<status>0</status>

<size unit="MB">120</size>

<reuse>false</reuse>

<autoExtend>true</autoExtend>

<increment unit="KB">10240</increment>

<maxSize unit="MB">-1</maxSize>

</DatafileAttributes>

<DatafileAttributes id="&lt;OMF\_SYSTEM\_DATAFILE\_0>">

<tablespace>SYSTEM</tablespace>

<temporary>false</temporary>

<online>true</online>

<status>0</status>

<size unit="MB">300</size>

<reuse>false</reuse>

<autoExtend>true</autoExtend>

<increment unit="KB">10240</increment>

<maxSize unit="MB">-1</maxSize>

</DatafileAttributes>

<DatafileAttributes id="&lt;OMF\_TEMP\_DATAFILE\_0>">

<tablespace>TEMP</tablespace>

<temporary>false</temporary>

<online>true</online>

<status>0</status>

<size unit="MB">20</size>

<reuse>false</reuse>

<autoExtend>true</autoExtend>

<increment unit="KB">640</increment>

<maxSize unit="MB">-1</maxSize>

</DatafileAttributes>

<DatafileAttributes id="&lt;OMF\_UNDOTBS1\_DATAFILE\_0>">

<tablespace>UNDOTBS1</tablespace>

<temporary>false</temporary>

<online>true</online>

<status>0</status>

<size unit="MB">200</size>

<reuse>false</reuse>

<autoExtend>true</autoExtend>

<increment unit="KB">5120</increment>

<maxSize unit="MB">-1</maxSize>

</DatafileAttributes>

<DatafileAttributes id="&lt;OMF\_UNDOTBS2\_DATAFILE\_0>">

<tablespace>UNDOTBS2</tablespace>

<temporary>false</temporary>

<online>true</online>

<status>0</status>

<size unit="MB">200</size>

<reuse>false</reuse>

<autoExtend>true</autoExtend>

<increment unit="KB">5120</increment>

<maxSize unit="MB">-1</maxSize>

</DatafileAttributes>

<DatafileAttributes id="&lt;OMF\_USERS\_DATAFILE\_0>">

<tablespace>USERS</tablespace>

<temporary>false</temporary>

<online>true</online>

<status>0</status>

<size unit="MB">5</size>

<reuse>false</reuse>

<autoExtend>true</autoExtend>

<increment unit="KB">1280</increment>

<maxSize unit="MB">-1</maxSize>

</DatafileAttributes>

<TablespaceAttributes id="SYSAUX">

<online>true</online>

<offlineMode>1</offlineMode>

<readOnly>false</readOnly>

<temporary>false</temporary>

<defaultTemp>false</defaultTemp>

<undo>false</undo>

<local>true</local>

<blockSize>-1</blockSize>

<allocation>1</allocation>

<uniAllocSize unit="KB">-1</uniAllocSize>

<initSize unit="KB">64</initSize>

<increment unit="KB">64</increment>

<incrementPercent>50</incrementPercent>

<minExtends>1</minExtends>

<maxExtends>4096</maxExtends>

<minExtendsSize unit="KB">64</minExtendsSize>

<logging>true</logging>

<recoverable>false</recoverable>

<maxFreeSpace>0</maxFreeSpace>

<autoSegmentMgmt>true</autoSegmentMgmt>

<bigfile>false</bigfile>

<datafilesList>

<TablespaceDatafileAttributes id="&lt;OMF\_SYSAUX\_DATAFILE\_0>">

<id>-1</id>

</TablespaceDatafileAttributes>

</datafilesList>

</TablespaceAttributes>

<TablespaceAttributes id="SYSTEM">

<online>true</online>

<offlineMode>1</offlineMode>

<readOnly>false</readOnly>

<temporary>false</temporary>

<defaultTemp>false</defaultTemp>

<undo>false</undo>

<local>true</local>

<blockSize>-1</blockSize>

<allocation>3</allocation>

<uniAllocSize unit="KB">-1</uniAllocSize>

<initSize unit="KB">64</initSize>

<increment unit="KB">64</increment>

<incrementPercent>50</incrementPercent>

<minExtends>1</minExtends>

<maxExtends>-1</maxExtends>

<minExtendsSize unit="KB">64</minExtendsSize>

<logging>true</logging>

<recoverable>false</recoverable>

<maxFreeSpace>0</maxFreeSpace>

<autoSegmentMgmt>true</autoSegmentMgmt>

<bigfile>false</bigfile>

<datafilesList>

<TablespaceDatafileAttributes id="&lt;OMF\_SYSTEM\_DATAFILE\_0>">

<id>-1</id>

</TablespaceDatafileAttributes>

</datafilesList>

</TablespaceAttributes>

<TablespaceAttributes id="TEMP">

<online>true</online>

<offlineMode>1</offlineMode>

<readOnly>false</readOnly>

<temporary>true</temporary>

<defaultTemp>true</defaultTemp>

<undo>false</undo>

<local>true</local>

<blockSize>-1</blockSize>

<allocation>1</allocation>

<uniAllocSize unit="KB">-1</uniAllocSize>

<initSize unit="KB">64</initSize>

<increment unit="KB">64</increment>

<incrementPercent>0</incrementPercent>

<minExtends>1</minExtends>

<maxExtends>0</maxExtends>

<minExtendsSize unit="KB">64</minExtendsSize>

<logging>true</logging>

<recoverable>false</recoverable>

<maxFreeSpace>0</maxFreeSpace>

<autoSegmentMgmt>true</autoSegmentMgmt>

<bigfile>false</bigfile>

<datafilesList>

<TablespaceDatafileAttributes id="&lt;OMF\_TEMP\_DATAFILE\_0>">

<id>-1</id>

</TablespaceDatafileAttributes>

</datafilesList>

</TablespaceAttributes>

<TablespaceAttributes id="UNDOTBS1">

<online>true</online>

<offlineMode>1</offlineMode>

<readOnly>false</readOnly>

<temporary>false</temporary>

<defaultTemp>false</defaultTemp>

<undo>true</undo>

<local>true</local>

<blockSize>-1</blockSize>

<allocation>1</allocation>

<uniAllocSize unit="KB">-1</uniAllocSize>

<initSize unit="KB">512</initSize>

<increment unit="KB">512</increment>

<incrementPercent>50</incrementPercent>

<minExtends>8</minExtends>

<maxExtends>4096</maxExtends>

<minExtendsSize unit="KB">512</minExtendsSize>

<logging>true</logging>

<recoverable>false</recoverable>

<maxFreeSpace>0</maxFreeSpace>

<autoSegmentMgmt>true</autoSegmentMgmt>

<bigfile>false</bigfile>

<datafilesList>

<TablespaceDatafileAttributes id="&lt;OMF\_UNDOTBS1\_DATAFILE\_0>">

<id>-1</id>

</TablespaceDatafileAttributes>

</datafilesList>

</TablespaceAttributes>

<TablespaceAttributes id="UNDOTBS2">

<online>true</online>

<offlineMode>1</offlineMode>

<readOnly>false</readOnly>

<temporary>false</temporary>

<defaultTemp>false</defaultTemp>

<undo>true</undo>

<local>true</local>

<blockSize>-1</blockSize>

<allocation>1</allocation>

<uniAllocSize unit="KB">-1</uniAllocSize>

<initSize unit="KB">512</initSize>

<increment unit="KB">512</increment>

<incrementPercent>50</incrementPercent>

<minExtends>8</minExtends>

<maxExtends>4096</maxExtends>

<minExtendsSize unit="KB">512</minExtendsSize>

<logging>true</logging>

<recoverable>false</recoverable>

<maxFreeSpace>0</maxFreeSpace>

<autoSegmentMgmt>true</autoSegmentMgmt>

<bigfile>false</bigfile>

<datafilesList>

<TablespaceDatafileAttributes id="&lt;OMF\_UNDOTBS2\_DATAFILE\_0>">

<id>-1</id>

</TablespaceDatafileAttributes>

</datafilesList>

</TablespaceAttributes>

<TablespaceAttributes id="USERS">

<online>true</online>

<offlineMode>1</offlineMode>

<readOnly>false</readOnly>

<temporary>false</temporary>

<defaultTemp>false</defaultTemp>

<undo>false</undo>

<local>true</local>

<blockSize>-1</blockSize>

<allocation>1</allocation>

<uniAllocSize unit="KB">-1</uniAllocSize>

<initSize unit="KB">128</initSize>

<increment unit="KB">128</increment>

<incrementPercent>0</incrementPercent>

<minExtends>1</minExtends>

<maxExtends>4096</maxExtends>

<minExtendsSize unit="KB">128</minExtendsSize>

<logging>true</logging>

<recoverable>false</recoverable>

<maxFreeSpace>0</maxFreeSpace>

<autoSegmentMgmt>true</autoSegmentMgmt>

<bigfile>false</bigfile>

<datafilesList>

<TablespaceDatafileAttributes id="&lt;OMF\_USERS\_DATAFILE\_0>">

<id>-1</id>

</TablespaceDatafileAttributes>

</datafilesList>

</TablespaceAttributes>

<RedoLogGroupAttributes id="1">

<reuse>false</reuse>

<fileSize unit="KB">51200</fileSize>

<Thread>1</Thread>

<member ordinal="0" memberName="redo\_01\_01\_01" filepath="+DATA1/{DB\_UNIQUE\_NAME}/"/>

</RedoLogGroupAttributes>

<RedoLogGroupAttributes id="2">

<reuse>false</reuse>

<fileSize unit="KB">51200</fileSize>

<Thread>1</Thread>

<member ordinal="0" memberName="redo\_01\_02\_01" filepath="+DATA1/{DB\_UNIQUE\_NAME}/"/>

</RedoLogGroupAttributes>

<RedoLogGroupAttributes id="3">

<reuse>false</reuse>

<fileSize unit="KB">51200</fileSize>

<Thread>2</Thread>

<member ordinal="0" memberName="redo\_02\_03\_01" filepath="+DATA1/{DB\_UNIQUE\_NAME}/"/>

</RedoLogGroupAttributes>

<RedoLogGroupAttributes id="4">

<reuse>false</reuse>

<fileSize unit="KB">51200</fileSize>

<Thread>2</Thread>

<member ordinal="0" memberName="redo\_02\_04\_01" filepath="+DATA1/{DB\_UNIQUE\_NAME}/"/>

</RedoLogGroupAttributes>

<RedoLogGroupAttributes id="5">

<reuse>false</reuse>

<fileSize unit="KB">51200</fileSize>

<Thread>1</Thread>

<member ordinal="0" memberName="redo\_01\_05\_01" filepath="+DATA1/{DB\_UNIQUE\_NAME}/"/>

</RedoLogGroupAttributes>

<RedoLogGroupAttributes id="6">

<reuse>false</reuse>

<fileSize unit="KB">51200</fileSize>

<Thread>2</Thread>

<member ordinal="0" memberName="redo\_02\_06\_01" filepath="+DATA1/{DB\_UNIQUE\_NAME}/"/>

</RedoLogGroupAttributes>

</StorageAttributes>

</DatabaseTemplate>

EOF

首先配置dbca.rsp文件，文件内容如下，将该文件保存为/tmp/dbca.rsp

cat > /tmp/dbca.rsp <<EOF

[GENERAL]

RESPONSEFILE\_VERSION = "10.0.0"

OPERATION\_TYPE = "createDatabase"

[CREATEDATABASE]

GDBNAME = "ORCL"

SID = "ORCL"

NODELIST="db01,db02"

TEMPLATENAME = "INST.dbt"

SYSPASSWORD = "1qaz2WSX"

SYSTEMPASSWORD = "1qaz2WSX"

EMCONFIGURATION = "NONE"

SYSMANPASSWORD = "1qaz2WSX"

DBSNMPPASSWORD = "1qaz2WSX"

STORAGETYPE=ASM

DISKGROUPNAME=DATA1

RECOVERYGROUPNAME=FRA1

CHARACTERSET = "AL32UTF8"

EOF

调用如下命令进行安装DATABASE SOFTWARE

export LANG=en\_US

dbca -silent -responseFile /tmp/dbca.rsp

## 更新数据库

注意：如下操作仅在一个节点进行即可。

cd $ORACLE\_HOME/rdbms/admin

sqlplus /nolog

SQL> CONNECT / AS SYSDBA

SQL> STARTUP

SQL> select \* from dba\_registry\_history;

SQL> @catbundle.sql psu apply

SQL> select \* from dba\_registry\_history;

SQL> @utlrp.sql

SQL> QUIT

检查如下文件，查看是否有报错信息

$ORACLE\_BASE/cfgtoollogs/catbundle/catbundle\_PSU\_<database SID>\_APPLY\_<TIMESTAMP>.log

$ORACLE\_BASE/cfgtoollogs/catbundle/catbundle\_PSU\_<database SID>\_GENERATE\_<TIMESTAMP>.log

# 安装后检查与配置

## 修改监听端口

RAC 10G修改监听端口，直接编辑listener.ora文件即可。在集群环境下，监听的名称一定要为LISTENER,ORACLE会自动在各个节点创建名称为LISTENER\_NODENAME的监听。

修改db local\_listener **(尽量使用IP地址，不要使用主机名)**

alter system set local\_listener = '(ADDRESS = (PROTOCOL = TCP)(HOST = db01-vip)(PORT = 1815))' scope=both sid='ORCL1';

alter system set local\_listener = '(ADDRESS = (PROTOCOL = TCP)(HOST = db02-vip)(PORT = 1815))' scope=both sid='ORCL2';

##+ASM1

alter system set local\_listener = '(ADDRESS = (PROTOCOL = TCP)(HOST = db01-vip)(PORT = 1815))' scope=both ;

###+ASM2

alter system set local\_listener = '(ADDRESS = (PROTOCOL = TCP)(HOST = db02-vip)(PORT = 1815))' scope=both;

## 备份文件

备份root.sh

官方建议在安装完成后要备份root.sh脚本文件。

cp /u01/app/oracle/product/10.2.0/crs\_1/root.sh /u01/backup/crs\_root.sh

cp /u01/app/oracle/product/10.2.0/crs\_1/install/root102.sh /u01/backup/crs\_root102.sh

cp /u01/app/oracle/product/10.2.0/db\_1/root.sh /u01/backup/oracle\_root.sh

备份spfile文件，集群环境下，下面命令慎用，集群会记录该信息。

$ sqlplus / as sysdba

SQL> create pfile='/u01/backup/db\_pfile.ora' from spfile;

## 检查集群状态

[oracle@db01 ~]$ crs\_stat -t

[oracle@db01 ~]$ ocrcheck

## 配置控制文件自动备份

每次数据文件备份时，都要求同时备份控制文件。可以进行如下设置：

configure controlfile autobackup on;

这样每次备份数据文件时会自动备份控制文件，其清理策略和数据文件清理策略相同。

备份控制文件会生成快照，以避免造成锁冲突，RAC时如果不明确设置该快照文件在ASM中，备份有时会报错ORA-00245，要求备份初始化时做如下设置：

CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+FRA1/snapcf\_orcl.f';

## 配置ASM参数

|  |  |  |
| --- | --- | --- |
| **PARAMETER** | **VALUE** | **DESC** |
| sga\_max\_size | 256M | alter system set sga\_max\_size=256M scope=spfile; |
| processes | 150 | Oracle推荐的计算公式为：25 + (10 + [max number of concurrent database file creations, and file extend operations possible])\*n，n为连接到ASM的实例个数。  alter system set processes=150 scope=spfile; |
|  |  |  |

## 配置DB参数

|  |  |  |
| --- | --- | --- |
| **PARAMETER** | **VALUE** | **DESC** |
| SGA\_TARGET | 24G |  |
| pga\_aggregate\_target | 8G | 使用该参数的前提是，必须WORKAREA\_SIZE\_POLICY=AUTO（默认值），并且数据库运行方式是Dedicate sever和连接时Dedicate方式时，这个参数才可生效。此时参数SORT\_AREA\_SIZE，HASH\_AREA\_SIZE，BITMAP\_AREA\_SIZE会被忽略失效。但如果使用MTS和SHARE SERVER连接方式(不推荐使用)时，则PGA\_AGGREGATE\_TARGET会被忽略失效，此时\*\_AREA\_SIZE就会生效。  PGA\_AGGREGATE\_TARGET是系统自动管理\*\_AREA\_SIZE的大小，因此推荐使用这个参数。 |
| lock\_sga | false | 为true时，会影响hugepage的使用。该参数不可以于AMM和ASMM同时使用。 |
| processes | 1000 |  |
| session\_cached\_cursors | 150 |  |
| DB\_BLOCK\_SIZE | 8K | OLTP类型的业务应用数据库系统，则DB\_BLOCK\_SIZE= 8k；  OLAP类型的业务应用数据库系统，则DB\_BLOCK\_SIZE= 32k；  在11.2.0.3实际应用过程中，发现凡是32K的实例，都出现了ORA-01628的错误，回滚段分配不能超过2G，因此在该问题解决前，要求全部设置为8K。 |
| OPEN\_CURSORS | 200 | 不小于200 |
| max\_dump\_file\_size | 100M | 以避免数据库多次dump出trc文件时迅速撑满整个文件系统。  注意该参数如果设置为纯数字，其单位为操作系统块数（每块512字节），因此应该严格设置为字符型的’100M’，而不是104857600。 |
| backup\_tape\_io\_slaves | true | 该参数设置为true后，通过模拟AIO的行为，主进程可以不必等待磁带写完成后就能继续处理其他数据块，会启动子进程来等待，从而加快了备份速度。 |
| filesystemio\_options | setall | 如果使用文件系统作为存储方式，建议设置该参数，以开启aio和direct io |

## 检查Hugepage是否使用正常

调用hugepages\_setting.sh验证当前hugepage设置是否正确。

使用grep Huge /proc/meminfo查看hugepage使用情况: total-free+resolved应接近SGA大小。



## 配置在线日志组

初始大小设置为500M。

alter database add logfile thread 1 group 5 ('+DATA1','+FRA1') size 500M ;

alter database add logfile thread 2 group 6 ('+DATA1','+FRA1') size 500M ;

alter database drop logfile group 1;

alter database add logfile thread 1 group 1 ('+DATA1','+FRA1') size 500M ;

alter database drop logfile group 3;

alter database add logfile thread 1 group 3 ('+DATA1','+FRA1') size 500M ;

alter database drop logfile group 2;

alter database add logfile thread 2 group 2 ('+DATA1','+FRA1') size 500M ;

alter database drop logfile group 4;

alter database add logfile thread 2 group 4 ('+DATA1','+FRA1') size 500M ;

## 配置归档模式

建议开启数据库归档模式。操作过程如下：

SQL> archive log list;

Database log mode No Archive Mode

Automatic archival Disabled

Archive destination USE\_DB\_RECOVERY\_FILE\_DEST

Oldest online log sequence 11

Current log sequence 12

SQL> shutdown immediate

Database closed.

Database dismounted.

ORACLE instance shut down.

SQL> startup mount

ORACLE instance started.

Total System Global Area 5044088832 bytes

Fixed Size 2261928 bytes

Variable Size 1090522200 bytes

Database Buffers 3909091328 bytes

Redo Buffers 42213376 bytes

Database mounted.

SQL> alter database archivelog;

Database altered.

SQL> alter database open ;

Database altered.

## 配置闪回数据库

flashback query（闪回查询）的功能可以使用户及时取得误操作前的数据，并可以针对错误进行相应的恢复措施。在默认参数UNDO\_MANAGEMENT = AUTO下 ，闪回查询就已经打开。但Flashback Database功能使用需要设置更多的参数，使用更多的存储空间。考虑到数据库系统的复杂性，使用Flashback Database来避免误操作可能带来的消极影响难以准确估计，因此不建议在生产库使用该功能。如果一定要用Flashback，建议使用基于表的闪回功能。

SQL> select flashback\_on from v$database;

FLASHBACK\_ON

------------------

NO

SQL> alter database flashback off;

Database altered.

## 编译相关模块

在建立或升级数据库后执行utlrp.sql脚本。这是一个可选操作，但oracle官方建议在安装后立即执行该脚本。

以oracle用户执行以下命令：

$ sqlplus /as sysdba

SQL> @$ORACLE\_HOME/rdbms/admin/utlrp.sql

## 配置UNDO/TMP表空间

SQL> select file#,name from v$datafile;

FILE# NAME

---------- ------------------------------------------------------------

1 +DATA1/orcl/datafile/system.256.938270497

2 +DATA1/orcl/datafile/sysaux.257.938270497

3 +DATA1/orcl/datafile/undotbs1.258.938270497

4 +DATA1/orcl/datafile/users.259.938270497

5 +DATA1/orcl/datafile/undotbs2.264.938270653

SQL> alter database datafile 3 autoextend off;

Database altered.

SQL> alter database datafile 3 resize 31G;

Database altered.

SQL> alter database datafile 5 autoextend off;

Database altered.

SQL> alter database datafile 5 resize 31G;

Database altered.

SQL> select file#,name from v$tempfile;

FILE# NAME

---------- ------------------------------------------------------------

1 +DATA1/orcl/tempfile/temp.263.938270585

SQL> alter database tempfile 1 autoextend off;

Database altered.

SQL> alter database tempfile 1 resize 31G;

Database altered.

## AWR报告默认文件名设置

为了使AWR报告默认生成的文件名更有可读性，要求编辑$ORACLE\_HOME/rdbms/admin/awrinpnm.sql。

第55行原来是

select '&&1'||:inst\_num||'\_'||:bid||'\_'||:eid||'&&2' dflt\_name from dual;

注释后修改为：

select '&&1'|| (select trim(INSTANCE\_NAME) from gv$instance where INSTANCE\_NUMBER =:inst\_num)||'\_'||(select to\_char(end\_interval\_time,'YYYYMMDD\_HH24MI') from dba\_hist\_snapshot where snap\_id=:bid and INSTANCE\_NUMBER =:inst\_num)||'\_'||(select to\_char(end\_interval\_time,'HH24MI') from dba\_hist\_snapshot where snap\_id=:eid and INSTANCE\_NUMBER =:inst\_num)||'&&2' dflt\_name from dual;

## 快照保存时间及间隔修改

快照保存时间从1周增加为40天，快照间隔从1小时增加为半小时：

exec dbms\_workload\_repository.modify\_snapshot\_settings(interval=>30,retention=>40\*24\*60);

确认办法：

SQL> select \* from dba\_hist\_wr\_control ;

## 收集统计信息

在用户对象创建前运行：

exec dbms\_stats.GATHER\_DATABASE\_STATS(no\_invalidate=>false);

exec dbms\_stats.GATHER\_FIXED\_OBJECTS\_STATS(no\_invalidate=>false);

收集系统信息，便于数据库引擎做出更贴近实际的IO、运行时间估算

exec dbms\_stats.GATHER\_SYSTEM\_STATS (gathering\_mode=>'NOWORKLOAD');

## 修改PROFILE

注意如下原则：

业务用户不得授予DBA角色；

坚持最小化特权原则。不得授予普通用户额外的全局权限，如select any/delete any/execute any等，应用有特殊需求的除外；

只有在对安全管理员进行相关权限和角色授予时，才允许带有关键字ADMIN OPTION；

除了sys、system、rman用户，其他oracle默认用户都应该被置为expire或lock状态，除非有特殊需求；

数据库所有活动状态用户的密码不得设置为已知的默认密码（如：sys用户密码也不能设置为manager）；

数据库用户的密码应该定期修改。

执行如下命令进行修改：

ALTER PROFILE DEFAULT limit FAILED\_LOGIN\_ATTEMPTS UNLIMITED PASSWORD\_LIFE\_TIME UNLIMITED;

## 与DRM有关的隐藏参数

10gRAC中引入的DRM（Dynamic Resource Mastering）的概念，在RAC的不同节点间根据资源使用情况决定master节点。由于10g中该功能仍然并不十分完善，有可能造成节点间master节点不断转换，触发bug，因此仍然推荐屏蔽该功能。屏蔽时需要配置如下隐藏参数（参数和11g不同）：

alter system set "\_gc\_affinity\_time"=0 scope=spfile;

alter system set "\_gc\_undo\_affinity"=FALSE scope=spfile;

## 块跟踪

块修改跟踪是Oracle 10g里推出的特性，会记录数据文件里每个块的更新信息，这些跟踪信息保存在跟踪文件里。 当启动块修改跟踪后，RMAN增量备份时使用跟踪文件里的信息，只读取上次0级备份以来有改变的块信息，而不用对整个数据进行扫描，从而提高了RMAN 备份的性能。

该技术默认禁用，如果启用增量备份，建议使用该技术。对OLAP大数据量系统，每次备份间隔中数据加载量小于总数据量1%的，可以考虑使用增量备份。

注意事项：

**在1次0级备份后，不宜超过8次增量备份。**

**块跟踪文件的大小可按照总大小的1/30000\*RAC节点数估算，默认建议设置为1G，可以满足两节点RAC 15T 的 备份需求。**

SQL> select \* from v$block\_change\_tracking;

SQL> col filename for a30

SQL> /

STATUS FILENAME BYTES

---------- ------------------------------ ----------

DISABLED

SQL> alter database enable block change tracking using file '+DATA1';

Database altered.

SQL> select status,bytes/1024/1024 from v$block\_change\_tracking ;

STATUS BYTES/1024/1024

---------- ---------------

ENABLED 11.0625

## 配置数据库自启动服务（单机）

数据库创建完成后，修改/etc/oratab，把对应数据库实例一行最后的N改成Y，然后调用dbstart看是否可以正常启动数据库实例。

为了方便管理，编写如下自启动脚本：

#!/bin/bash

#start and stop the oracle instance

# chkconfig -level 345 --add oradb

#chkconfig: 345 91 19

# description: starts the oracle listener and instance

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=$ORACLE\_BASE/product/10.2.0/db\_1

export PATH=$ORACLE\_HOME/bin:$PATH

export ORACLE\_OWNER="oracle"

if [ ! -f $ORACLE\_HOME/bin/dbstart -o ! -d $ORACLE\_HOME ]

then

echo "oracle startup:cannot start"

exit 1

fi

case "" in

start)

#startup the listener and instance

echo -n "oracle startup: "

su - $ORACLE\_OWNER -c "$ORACLE\_HOME/bin/lsnrctl start"

su - $ORACLE\_OWNER -c $ORACLE\_HOME/bin/dbstart

touch /var/lock/subsys/oracle

echo "finished"

;;

stop)

# stop listener, apache and database

echo -n "oracle shutdown:"

su - $ORACLE\_OWNER -c "$ORACLE\_HOME/bin/lsnrctl stop"

su - $ORACLE\_OWNER -c $ORACLE\_HOME/bin/dbshut

rm -f /var/lock/subsys/oracle

echo "finished"

;;

reload|restart)

stop

start

;;

\*)

echo "Usage: oradb [start|stop|reload|restart]"

exit 1

esac

exit 0

脚本存为/etc/rc.d/init.d/oradb后，调用如下命令，添加自启动服务。

chmod a+x /etc/rc.d/init.d/oradb

chkconfig --add oradb

chkconfig on oradb

## 数据库合规及健康检查

对数据库进行合规和健康检查。

# 卸载ORACLE 10G RAC

## 卸载RAC

## 删除oracle自动安装的文件，删除 /etc下ora开头的文件或目录

rm -rf /etc/ora\*

rm -rf /tmp/.oracle

rm -rf /var/tmp/.oracle

rm -rf /home/oracle/.oracle

rm -rf /home/oracle/oradiag\*

删除/etc/inittab.crs，并用/etc/inittab.no\_crs覆盖/etc/inittab（可选）

rm -rf /etc/oracle/

rm -f /etc/init.d/init.cssd

rm -f /etc/init.d/init.crs

rm -f /etc/init.d/init.crsd

rm -f /etc/init.d/init.evmd

rm -rf /etc/init.d/\*ohasd\*

cp /etc/inittab.orig /etc/inittab

rm -f /etc/rc2.d/\*nit.cssd

rm -f /etc/rc2.d/\*nit.crs

rm -f /etc/rc2.d/\*nit.crsd

rm -f /etc/rc2.d/\*nit.evmd

rm -f /etc/rc2.d/\*ohasd\*

rm -f /etc/rc3.d/\*nit.cssd

rm -f /etc/rc3.d/\*nit.crs

rm -f /etc/rc3.d/\*nit.crsd

rm -f /etc/rc3.d/\*nit.evmd

rm -f /etc/rc3.d/\*ohasd\*

rm -f /etc/rc4.d/\*nit.cssd

rm -f /etc/rc4.d/\*nit.crs

rm -f /etc/rc4.d/\*nit.crsd

rm -f /etc/rc4.d/\*nit.evmd

rm -f /etc/rc4.d/\*ohasd\*

rm -f /etc/rc5.d/\*nit.cssd

rm -f /etc/rc5.d/\*nit.crs

rm -f /etc/rc5.d/\*nit.crsd

rm -f /etc/rc5.d/\*nit.evmd

rm -f /etc/rc5.d/\*ohasd\*

rm -rf /tmp/CVU\_\*

rm -rf /tmp/OraInsta\*

rm -rf /opt/ORCLfmap

rm -rf /usr/local/bin/dbhome

rm -rf /usr/local/bin/oraenv

rm -rf /usr/local/bin/coraen

\*\*\*\*\*\*\*\*\*\*\*\*格式化相关磁盘\*\*\*\*\*\*\*\*\*\*\*\*\*\*

dd if=/dev/zero of=/dev/raw/raw1 bs=1024 count=5000

dd if=/dev/zero of=/dev/raw/raw2 bs=1024 count=5000

dd if=/dev/zero of=/dev/raw/raw3 bs=1024 count=5000

dd if=/dev/zero of=/dev/raw/raw4 bs=1024 count=5000

dd if=/dev/zero of=/dev/raw/raw5 bs=1024 count=5000

dd if=/dev/zero of=/dev/mapper/RACDATA1p1 bs=1024 count=50000

dd if=/dev/zero of=/dev/mapper/RACFRA1p1 bs=1024 count=50000

## 回退root.sh

参照文章《443418.1》，具体步骤如下：

crsctl stop crs

/u01/app/oracle/product/10.2.0/crs\_1/install/rootdelete.sh

/u01/app/oracle/product/10.2.0/crs\_1/install/rootdeinstall.sh

rm -rf /tmp/.oracle

rm -rf /var/tmp/.oracle

# 云实施的支持

## 基础设施云变化对数据库的影响

新一代建设中，操作系统将提供云服务，数据库服务器可以动态扩充CPU、内存。对动态增加的CPU，数据库后台进程、连接进程都可以立即使用该资源，仅仅是数据库根据CPU个数自动计算的一些内部资源限制，如latch数等，需要待实例重启后才能重新计算。对动态增加的内存，SGA无法利用，因为SGA\_TARGET等参数已经做了限定，无法自动扩大；但PGA、连接进程仍然可以自动使用这部分内存。CPU、内存的动态降低影响难以评估，建议不这样操作。

## 数据库池

为了提高资源利用率，降低数据中心管理的实例数，避免数据库烟囱式、竖井式的传统使用方式，需要引入数据库池的概念。数据库池还可以提高对应用数据库使用需求的响应速度，加快系统上线速度，同时有利于规范化的安装管理。

数据库池的实现允许在一个数据库实例中部署属于不同应用的数据，不同应用的数据属于不同的用户（Schema），因此需要对权限进行更严格的控制，尤其必须禁止对应用户授予DBA权限。为了避免不同用户间对资源的争用，特别是某个应用占用全部资源导致其他应用都受到影响，需要使用Oracle提供的Resource Manager工具，该工具可以协助DBA配置合适的策略，限定各个用户对资源的使用，避免上述现象的出现。

# 故障解决

## java: error while loading shared libraries:libpthread.so.0: cannot open shared object file: No such file or directory

没有按照4.2节修改vipca和srvctl文件。

## The given interface(s), "eth0" is not public. Public interfaces should be used to configure virtual IPs.

图形界面上运行$CRS\_HOME/bin/vipca，手工重新配置rac1-vip和rac2-vip。

# /u01/oracle/product/10.2.0/crs\_1/bin/vipca

## Error 0(Native: listNetInterfaces:[3])[Error 0(Native: listNetInterfaces:[3])]

[root@rac2 bin]# ./oifcfg iflist

eth0 10.1.3.0

eth1 10.10.10.0

[root@rac2 bin]# ./oifcfg setif -global eth0/10.1.3.0:public

[root@rac2 bin]# ./oifcfg setif -global eth1/10.10.10.0:cluster\_interconnect

[root@rac2 bin]# ./oifcfg getif

eth0 192.168.100.0 global public

eth1 10.10.10.0 global cluster\_interconnect

然后在视窗界面重新执行vipca即可。

## 操作系统不支持

开始之前需要修改 oraparam.ini 文件，否则，会报操作系统不被支持。把检测操作系统那一步注上:

[Certified Versions]

Linux=redhat-3,SuSE-9,redhat-4,UnitedLinux-1.0,asianux-1,asianux-2

或者

通过创建一个新文件/etc/redhat-release并添加以下行，欺骗安装程序让它以为我们  
的系统是RedHat:Red Hat Linux release 4.1

## ins\_emdb.mk问题

安装过程中，接近安装进度84%左右时出现错误警告:error in invoking target ‘collector’ of makefile ‘/home/oracle/product/10.2.0/db\_1/sysman/lib/ins\_emdb.mk’

解决办法1：

忽略后，数据库软件安装成功。升级安装补丁后会修正该错误，该错误并不影响程序的正常运行。

解决办法2：

另有解决方法如下：修改oracle可执行文件，在执行该文件前设置环境变量DISABLE\_HUGETLBFS=1

## ORA-27125

数据库创建过程中出现如下错误：

ORA-27125: unable to create shared memory segment" when setting up the database

解决方法如下：将oinstal组的id存放到/etc/sys/vm/hugetlb\_shm\_group

## 无法启动 ONS 资源

当 VIPCA 尝试启动 ONS 应用程序资源时，出现错误消息“CRS-0215:Could not start resource ‘ora.db02.ons’”。

解决方法：

ONS 尝试访问本地主机，但无法解析 IP 地址。将以下项添加到 /etc/hosts 中。

127.0.0.1 localhost