Skillset4

Skillset4

rac的考察，他们在题目中提供了下面的这些内容，这些内容我们是可以直接原封不动的填到gi安装的那个部分的，一会儿到那个部分的时候我再给大家看下

就是这部分的内容，不过这儿名字跟考试的可能不大一样，记不太清了，反正到时候考试的时候照着那个进行填写就行了

Cluster Name: cluster

Scan Name: cluster-scan.cluster.example.com

Scan Port: 1521

GNS VIP Address : 192.168.157.140

GNS Sub Domain : cluster.example.com

剩余的错误忽视掉，不影响安装

Section 1：安装GI和创建ASM磁盘组

1.使用grid用户安装如下要求安装GI

(1). grid用户的BASE目录为/u01/app/grid，家目录为/u01/app/12.1.0/grid

(2). 创建OCR和Voting Disk，使用+DGDATA

(3). 集群可以支持不同角色的节点

(4). host01和host02必须能够访问ASM实例

(5). 单个ASM实例的失败，不会导致任意节点数据库实例的失败

(6). 心跳线要求高可用性

2.创建一个磁盘组+DGDATA，使用normal冗余，使用如下asmdisks

/dev/asmdisk1p1

/dev/asmdisk1p2

/dev/asmdisk1p3

/dev/asmdisk1p4

/dev/asmdisk2p1

/dev/asmdisk2p2

3.创建一个磁盘组+DGACFS，使用external冗余，使它可以用于ADVM卷，使用如下asmdisks

/dev/asmdisk1p5

/dev/asmdisk1p6

4.创建一个磁盘组+DGFRA，使用external冗余，用于快速恢复区，使用如下asmdisks

/dev/asmdisk1p7

/dev/asmdisk1p8

5.使用ADVM卷，在ACFS磁盘组中创建一个ACFS文件系统，名称为ACFS_VOL1，使用如下描述

大小：2G

挂载点：/u01/app/oracle/acfs_share

Section 2：管理GI

1.在DGFRA和DGACFS磁盘组上镜像OCR。

[root@host01 ~]# cd /u01/app/12.1.0/grid/bin/

```
[root@host01 bin]# ./ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       2820
         Available space (kbytes) :     259300
         ID                       : 238905617
         Device/File Name         :      +DGDATA
                                    Device/File integrity check succeeded

                                    Device/File not configured

                                    Device/File not configured

                                    Device/File not configured

                                    Device/File not configured

         Cluster registry integrity check succeeded

         Logical corruption check succeeded


[root@host01 bin]# ./ocrconfig -add +DGFRA
[root@host01 bin]# ./ocrconfig -add +DGACFS
[root@host01 bin]# ./ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       2820
         Available space (kbytes) :     259300
         ID                       : 238905617
         Device/File Name         :      +DATA
                                    Device/File integrity check succeeded
         Device/File Name         :      +DGFRA
                                    Device/File integrity check succeeded

                                    Device/File not configured

                                    Device/File not configured

                                    Device/File not configured

         Cluster registry integrity check succeeded

         Logical corruption check succeeded

[root@host02 ~]# /u01/app/11.2.0/grid/bin/ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       2820
```

Available space (kbytes) : 259300

ID : 238905617

Device/File Name : +DATA

Device/File integrity check succeeded

Device/File Name : +FRA

Device/File integrity check succeeded

Device/File not configured

Device/File not configured

Device/File not configured

Cluster registry integrity check succeeded

Logical corruption check succeeded

2.在host01上dump OLR文件，在/home/oracle/目录中创建ASMOLR文件。

1）做host01的OLR的dump

[oracle@host01 acfs]$ su -

Password:

[root@host01 ~]# cd /u01/app/11.2.0/grid/bin/

[root@host01 bin]# ./ocrconfig -local -export /home/oracle/ASMOLR

[root@host01 bin]# cd /home/oracle

[root@host01 oracle]# ls

db_images.zip MYOLR oradiag_oracle

[root@host01 oracle]#

2）做host02的OLR的dump

[oracle@host02 ~]$ su -

Password:

[root@host02 ~]# cd /u01/app/11.2.0/grid/bin/

[root@host02 bin]# ./ocrconfig -local -export /home/oracle/MYOLR

[root@host02 bin]# cd /home/oracle

[root@host02 oracle]# ls

MYOLR oradiag_oracle

[root@host02 oracle]#

3.关闭GI自启动。

1）关闭host01的自动启动功能

[root@host01 ~]# cd /u01/app/11.2.0/grid/bin/

[root@host01 bin]# ./crsctl disable has

CRS-4621: Oracle High Availability Services autostart is disabled.

2）关闭host02的自动启动功能

[root@host02 ~]# cd /u01/app/11.2.0/grid/bin/

[root@host02 bin]# ./crsctl disable has

CRS-4621: Oracle High Availability Services autostart is disabled.

4.复制host01目录/home/oracle/scripts/pictures.zip文件，到ACFS文件系统/u01/app/oracle/acfs_share

2）查看ACFS文件系统信息

```
[13:41:00 root(grid)@rac1 ~]# acfsutil info fs
/u01/app/oracle/acfs_share
    ACFS Version: 12.1.0.2.0
    on-disk version:       39.0
    flags:        MountPoint,Available
    mount time:   Fri Nov 13 13:20:37 2015
    allocation unit:       4096
    volumes:      1
    total size:   9663676416  (    9.00 GB )
    total free:   9497395200  (    8.84 GB )
    file entry table allocation: 49152
    primary volume: /dev/asm/acfs_vol1-472
        label:
        state:                 Available
        major, minor:          251, 241665
        size:                  9663676416  (    9.00 GB )
        free:                  9497395200  (    8.84 GB )
        ADVM diskgroup         DGACFS
        ADVM resize increment: 67108864
        ADVM redundancy:       unprotected
        ADVM stripe columns:   8
        ADVM stripe width:     1048576
    number of snapshots:  0
    snapshot space usage: 0  ( 0.00 )
    replication status: DISABLED
```

3)查看已经挂载的ACFS文件系统

```
[root@host01 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             6.8G  3.4G  3.1G  53% /
/dev/sda3             4.9G  4.4G  274M  95% /stage
/dev/sda2              15G  3.7G   11G  27% /u01
tmpfs                 506M  150M  356M  30% /dev/shm
/dev/asm/db_files-260
                      2.0G   73M  2.0G   4% /u01/app/oracle/acfs
```

注:如果没有自动自动挂载,可以在root下执行

```
[root@host01 ~]# cd sbin
[root@host01 sbin]# /sbin/mount.acfs -o all
```

4)在host02上验证能否查看到db_images.zip文件

```
[root@host02 ~]# su - oracle
[oracle@host02 ~]$ cd /u01/app/oracle/acfs_share
[oracle@host02 acfs]$ ls
pictures.zip  lost+found
[oracle@host02 acfs_share]$
```

5.创建ACFS文件系统快照,名称为SNAP01

1)给ACFS做sanpshot,并在host01上验证

```
[oracle@host01 ~]$ /sbin/acfsutil snap create SNAP01 /u01/app/oracle/acfs/
acfsutil snap create: Snapshot operation is complete.
```

[oracle@host01 ~]$ ls /u01/app/oracle/acfs/.ACFS/snaps
ACFS_SNAP
[oracle@host01 snaps]$ cd ACFS_SNAP/l
[oracle@host01 ACFS_SNAP]$ ls
db_images.zip lost+found
2)在host02上验证
[oracle@host02 ~]$ cd /u01/app/oracle/acfs/.ACFS/snaps
[oracle@host02 snaps]$ ls
ACFS_SNAP
[oracle@host02 snaps]$ cd ACFS_SNAP/
[oracle@host02 ACFS_SNAP]$ ls
db_images.zip lost+found
注:snapshot删除的方法如下:
[oracle@host01 ~]$ /sbin/acfsutil snap delete ACFS_SNAP /u01/app/oracle/acfs/
acfsutil snap delete: Snapshot operation is complete.
[oracle@host01 ~]$ ls /u01/app/oracle/acfs/.ACFS/snaps

6.配置OCR的备份位置,路径为/home/oracle/ocrbackup
[root@host01 bin]# ./ocrconfig -showbackup

host01 2015/10/12 21:57:14 /u01/app/12.1.0/grid/cdata/host-cluster/backup00.ocr 0

host01 2015/10/12 17:57:13 /u01/app/12.1.0/grid/cdata/host-cluster/backup01.ocr 0

host01 2015/10/12 13:57:11 /u01/app/12.1.0/grid/cdata/host-cluster/backup02.ocr 0

host01 2015/10/12 13:57:11 /u01/app/12.1.0/grid/cdata/host-cluster/day.ocr 0

host01 2015/10/12 13:57:11 /u01/app/12.1.0/grid/cdata/host-cluster/week.ocr 0
PROT-25: Manual backups for the Oracle Cluster Registry are not available

[oracle@host01 ~]$ mkdir ocrbackup

[root@host01 bin]# ./ocrconfig -backuploc /home/oracle/ocrbackup

7.配置集群策略管理方法

这一部分,我们一定需要注意,这儿所有的策略的步骤,我们一定是要用grid用户去进行操作,不然,因为设置policy的时候会自动的生成server pool,如果我们用root用户执行的命令,那么生成的pool,oracle用户是没有权限使用这个pool的,自然也就没办法创建数据库了,这儿我们一定要注意!!!!

[14:08:55 root(grid)@rac1 bin]# crsctl status server rac1 -f
NAME=rac1
MEMORY_SIZE=6018
CPU_COUNT=2
CPU_CLOCK_RATE=2394

CPU_HYPERTHREADING=1
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=Free
STATE_DETAILS=
ACTIVE_CSS_ROLE=hub

[14:09:17 root(grid)@rac1 bin]# crsctl status server rac2 -f
NAME=rac2
MEMORY_SIZE=3940
CPU_COUNT=2
CPU_CLOCK_RATE=2394
CPU_HYPERTHREADING=1
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=Free
STATE_DETAILS=
ACTIVE_CSS_ROLE=hub

[14:09:32 root(grid)@rac1 bin]# crsctl status category
NAME=ora.hub.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=

NAME=ora.leaf.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=leaf
EXPRESSION=

[14:10:11 root(grid)@rac1 bin]# crsctl status server -category ora.hub.category
NAME=rac1
STATE=ONLINE

NAME=rac2
STATE=ONLINE

检查/home/oracle/scripts/policysetfile.txt文件，执行必要的操作让这个配置文件可以被加载。创建的serverpool将用于后面的数据库创建。
激活day policy

SERVER_POOL_NAMES=Free bigpool ora.racdbpool

POLICY
NAME=day
DESCRIPTION=The day policy
SERVERPOOL
NAME=ora.racdbpool
IMPORTANCE=10
MAX_SIZE=5
MIN_SIZE=1
SERVER_CATEGORY=ora.hub.category
SERVERPOOL
NAME=bigpool
IMPORTANCE=0
MAX_SIZE=0
MIN_SIZE=0
SERVER_CATEGORY=big

POLICY
NAME=night
DESCRIPTION=The night policy
SERVERPOOL
NAME=ora.racdbpool
IMPORTANCE=10
MAX_SIZE=5
MIN_SIZE=1
SERVER_CATEGORY=ora.hub.category
SERVERPOOL
NAME=bigpool
IMPORTANCE=5
MAX_SIZE=1
MIN_SIZE=1
SERVER_CATEGORY=big

srvctl status srvpool -serverpool Free -detail

[oracle@host01 bin]$ ./crsctl add category big
[oracle@host01 bin]$ ./crsctl modify policyset -file /home/oracle/policysetfile.txt
[oracle@host01 bin]$ ./crsctl status policyset
[oracle@host01 bin]$ ./crsctl modify policyset -attr "LAST_ACTIVATED_POLICY=night"

下面开始安装数据库的软件，这儿我需要说的是
我们在安装完gi以后就需要直接去安装数据库软件了，在安装数据库软件的时候再去做上面gi相关的配置的工作，不然的话，我们的考试时间肯定是不够的，我们需要并行的安排考试的时间

Section 3：安装数据库软件和创建RAC
1.使用oracle用户安装数据库软件，BASE目录为/u01/app/oracle,HOME目录为/u01/app/oracle/product/12.1.0/dbhome_1

2.创建RAC

3.使用general purpose, policy-manage管理, 数据库名RACDB。不配置EM

4.使用racdbpool作为server pool

5.sys和system用户口令必须为oracle

6.使用+DGDATA磁盘组, 用来存放所有数据库文件

7.在+DGFRA中配置快速恢复区, 大小为4G

8.不要安装sample

9.内存设置为1G

Section 4: 配置RAC环境

1.创建序列, 名称为PS_SEQ。序列在所有节点中要有序, 要在内存中维护。

开始值:100

结束值:99999

ORACLE RAC环境下读取序列乱序问题 分类: DataBase 2013-02-21 12:59 2986人阅读 评论(0) 收藏 举报

在数据库部署了RAC环境之后, 偶尔会出现从Oracle Sequence所取出来的数是混乱的, 比如第二次比第一次所取的数要小。这样当程序的逻辑依赖于ID的大小来排序时, 就会产生系统混乱。

其实问题是出在数据库是个RAC环境, 序列是被共享的, 序列默认是有缓存的。假设RAC上的两个节点上序列缓存设为20, 第一个节点上缓存1-20, 第二个节点缓存了21-40, 当从不同节点来进行对sequence取值的时候, 从第二个节点上取的值就会比从第一个节点上取的要大。而且默认序列都是noorder的。因为很有可能出现这种情况。

具体方法有两个:

1. 设置cache为空

2. 创建序列的时候设置为order, 即采用cache + order

sequence创建方法:

```
CREATE SEQUENCE [schema.]sequence
    [INCREMENT BY integer]
    [START WITH integer]
    [MAXVALUE integer | NOMAXVALUE]
    [MINVALUE integer | NOMINVALUE]
    [CYCLE | NOCYCLE]
    [CACHE integer | NOCACHE]
    [ORDER | NOORDER]
```

Oracle下关于Sequence的使用有三种情况:

1. cache + noorder

2. nocache

3. cache + order

Oracle为了管理sequence使用了以下三中锁

(1)row cache lock

在调用sequence.nextval过程中，将数据字典信息进行物理修改时获取，赋予了nocache属性的sequence上发生。

(2)SQ锁 -- enq: SQ

在内存上缓存(cache)范围内，调用sequence.nextval期间拥有此锁，赋予了cache+noorder 属性的sequence上发生。

(3)SV锁 -- DFS lock handle

RAC上节点之间顺序得到保障的情况下，调用sequence.nextval期间获得，赋予了cache+order属性的sequence上发生。

赋予了CACHE属性的sequence调用nextval期间，应该以SSX模式获得SQ锁，许多会话同时为了获取SQ锁而发生争用过程中，若发生争用，则等待enq:SQ-contention.

enq:SQ-contention事件的P2值是sequence的object ID,因此，若利用P2值与DBA_OBJECTS的结合，就可以知道对哪个、Sequence发生了等待对象。

创建Sequence赋予的CACHE值较小时，有enq:SQ-contention等待增加的趋势，CACHE值较小，内存上事先CACHE的值很快被耗尽，这时需要将数据字典信息物理修改，再次执行CACHE的工作，在此期间，因为一直要拥有SQ锁，相应的Enq:SQ-contention事件的等待时间也会延长，很不幸的是，在创建Sequence时，将CACHE值的缺省值设定为较小20,因此创建使用量最多的Sequence时，CACHE值应该取1000以上的较大值。

偶而一次性同时创建许多会话，有时会发生enq:SQ-contention等待事件，其理由是V$SESSION.AUDSID(auditing sessionid) 列值是利用Sequence创建的，oracle在创建新的会话后，利用名为SYS.AUDSESS$的sequence的nextval创建AUDSID的值，SYS.AUDSESS$ Sequence的CACHE大小的缺省值设定为 20，许多会话同时连接，可以将SYS.AUDSESS$ sequence的CACHE大小扩大至1000，以此可以解决 enq:SQ-contention等待问题。

RAC上创建Sequence时，在赋予了CACHE属性的状态下:

(1)若没有赋予ORDER属性，则各节点将会把不同范围的Sequence值CACHE到内存上，比如拥有两个节点的RAC环境下，创建CACHE值为100的 sequence时，1节点会使用1-100，2节点会使用101-200。使用时从各自节点取sequence。

(2)若两个节点之间会通过递增的使用sequence, 必须赋予如下ORDER属性。

SQL>Create sequence ordered_sequence cache 100 order;

在order 的情况下, 2个节点取的sequence是递增的。下文会有示例来说明这两种情况。

如果已赋予CACHE+ORDER属性的sequence, oracle使用SV锁进行行同步, 即, 对赋予了ORDER属性的sequence调用nextval时, 应该以SSX模式拥有SV锁, 在获取SV锁过程中, 若发生了争用, 不是等待ROW CACHE或者是enq:SQ-contention,而是等待名为DFS lock handle事件, 正因如此V$EVENT_NAME视图上不存在类似与"enq:SV-contention"

DFS lock handle事件是在OPS或者RAC环境下, 除了 高速缓冲区 同步之外, 还有 行高速缓冲区 或者 库高速缓冲区 同步获取锁的过程中的等待事件。 若保证全局范围内获得锁, 在此过程中会发生DFS look handle等待, 在获取SV锁的过程中发生的DFS lock handle等待事件的P1,P2值与enq:SQ-contention等待事件相同(p1=mode+namespace,p2=object#).因此会从P1值能确认是否是SV锁, 通过P2可以确认哪些是Sequence发生过等待.

SV锁争用问题发生时的解决办法与SQ锁的情况相同, 就是CACHE值进行适当的调整, 这也是唯一的方法。

2.在racdbpool中创建和激活一个service , 名称为RACDBACN, 在任意时间只允许登录一个实例
srvctl add service -db racdb -service RACDBACN -serverpool racdbpool -cardinality SINGLETON

3.在racdbpool中创建和激活一个service , 名称为ACNTRECV, 支持事务级的故障转移。
srvctl add service -db racdb -service ACNTRECV -serverpool racdbpool -failovertype TRANSACTION -commit_outcome TRUE

4.转换RAC到RAC one node
srvctl convert database -db racdb -dbtype RACONENODE
  [-instance instance_name -timeout timeout]
-w timeout]

5.SALES表使用序列SALSEQ

6.转换one node到RAC
srvctl convert database -db racdb -dbtype RAC [-node node_name]

7.配置RACDB, 减少并行查询重复块镜像。当重复的块镜像存在多个实例的buffer cache中。
alter system set parallel_force_local=true sid='*';

8.配置RACDB控制文件自动备份, 无论任何实例导致的数据库物理结构改变。