Goodus 기술노트 [36 회]
11g R1 Clusterware 를 이용한 Oracle H/A 구성

Author	**유광복,조연철,백순성**
Creation Date	**2008-11-31**
Last Updated	**2008-12-03**
Version	**1.0**

Version	변경일자	변경자(작성자)	주요내용
1	2008-12-03	유광복,조연철,백순성	문서 최초 작성
2			
3			

Goodus
Breathing Life into Technology

Contents

1.1. Oracle Clusterware

1.1.1. Oracle Clusterware 기본 동작

- kill -9 'LGWR Process' ⇒ Restart Automatically

  kill -9 'listener PID' ⇒ Restart Automatically
- kill 'oracle.exe' by Windows Task Manager ⇒ Restart Automatically

  kill 'tnslsnr.exe' by Windows Task Manager ⇒ Restart Automatically
- When a Node dies ⇒ VIP fails over to a different node

1.1.2. 10g Release 2 is Protect 3rd party Applications

Oracle Clusterware 를 사용하여 Single Instance 환경의 HA 기능도 제공하며, Single Node / Local FileSystem 을 ASM 또는 Cluster FileSystem 인 OCFS 등으로 전환하여 High Availability 기능을 이용할 수 있다.

10g R1 까지 지원되지 않던 Non-Oracle Application 의 Failure Protect 기능이 10g R2 부터 가능해졌으므로 본 문서는 위의 기능을 기반으로 Test 하였다.

1.2. The H/A Framework

오라클에서 제공하는 고가용성 솔루션은 일반적으로 RAC 를 의미한다. 이는 최소 두개 System 을 기반으로 1 개의 Databaes 를 노드 개수만큼의 인스턴스에서 서비스를 할 수 있는 장점을 가지고 있다. 즉 특정 시스템에서 장애가 발생해도 남아 있는 시스템에서 서비스를 계속 수행하기 때문에 무중단 서비스를 구현이 가능하다.

H/A 기반은 Active – Standby 구조로 고가용성 솔루션을 제공한다. 9i 에서는 OS Vender Cluster 를 기반으로만 구현이 가능하였는데 10g Clusterware 에서 그 기능을 완벽하게 수행할 수 있다. 아래 문서는 서로 다른 두 개의 Database 구성하여 각각 Active – Standby Database 를 구성을 하고 실제 failover 발생시 어떻게 지속적으로 서비스 수행이 가능한지를 구현하였다. H/A 는 RAC 에 비해서 system fault 가 발생하면 service down 이 조금 발생함을 알려 드립니다.

1.2.1. Shared Resources

- A common disks

: OCR & Vote for Oracle clusterware system files

- A common Network interconnect

- Same OS

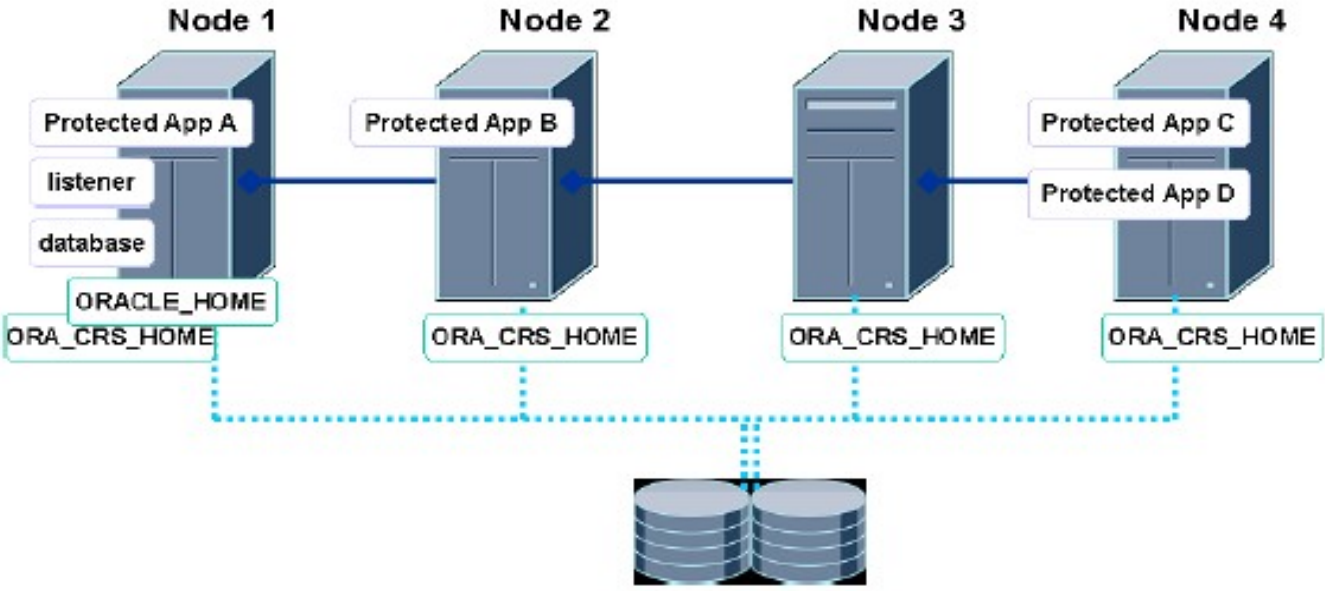
1.2.2. 4 Node configurations

- Node 1 : Running Oracle Instance & Clusterware, Protecting Application A

- Node 2 : Running Oracle Clusterware, Protecting Application B

- Node 3 : Running Oracle Clusterware

- Node 4 : Running Oracle Clusterware, Protecting Application C & D



1.2.3. Oracle H/A 기본 구성 절차

Hardware 구성 및 OS 구성은 10g RAC 설치와 동일하게 이뤄 지며 Vendor Clusterware 없이 구현이 가능합니다.

1) install Oracle S/W Only into a new Oracle Clusterware HOME on the 2nd Node

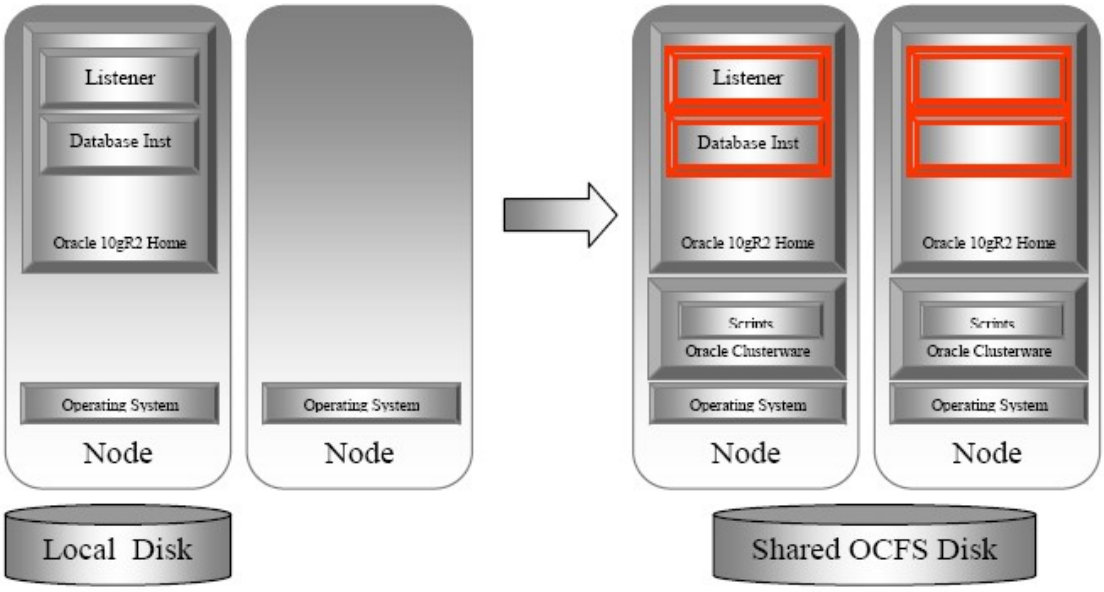
2) Install Cluster Filesystem both nodes

3) Modify Oracle Configurations

- orapwd
- init<SID>.ora
- listener.ora, tnsnames.ora

4) Create / Tegister / Start / Relocate Resources

- $ORA_CRS_HOME/crs/public



1.2.4. Using the Application Framework

1) Create an VIP

- required if the application is access via network clients

2) Create an Action Program

- Oracle Clusterware to start, stop & query the status check

- C, java or 기타 scripting language

- http://otn.oracle/com/rac => RAC sample code =>

> Sample HA Agent for Oracle Database [Nov 2006]
> These scripts should be used in conjunction with the white paper "Using Oracle Clusterware to Protect a Single Instance Oracle Database" available from the main Oracle RAC web site and included in this zip file. They allow you to use Oracle Clusterware to protect a Single Instance Oracle Database and optionally an ASM instance and Listener.

3) Create an Application Profile

- Describes the process and limits to protect

4) Register the Application

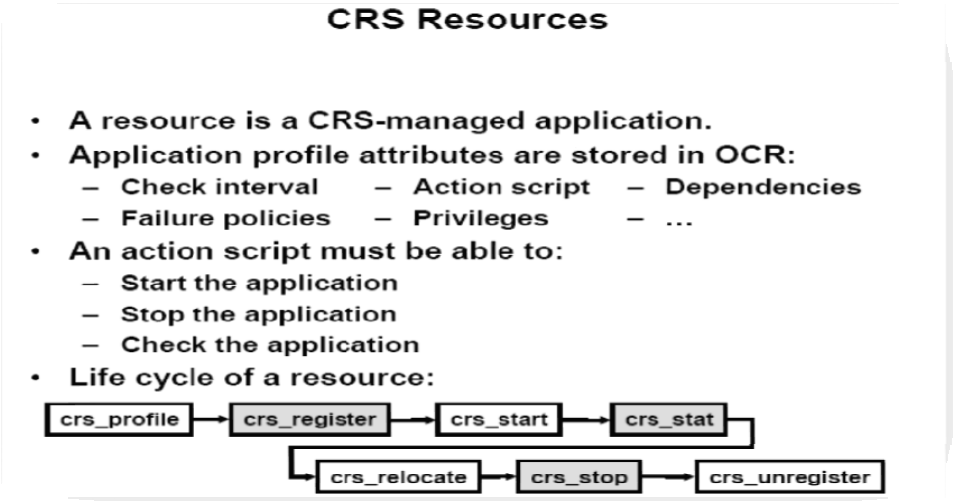
- Register the Application Profile with Oracle Clusterware

5) Set the permission on your Application

1.3. Oracle H/A 구성

1.3.1. 환경 설정

	Node 1	Node 2
Hostname	goodus1	goodus2
/etc/hosts	61.250.99.232 goodus1 loghost 61.250.99.233 goodus2 61.250.99.234 goodus1-vip 61.250.99.236 goodus2-vip 172.16.0.129 goodus1-priv 172.16.0.130 goodus2-priv	61.250.99.233 goodus2 loghost 61.250.99.232 goodus1 61.250.99.234 goodus1-vip 61.250.99.236 goodus2-vip 172.16.0.129 goodus1-priv 172.16.0.130 goodus2-priv
CRS, RDBMS user	oracle, dba	oracle, dba
CRS_HOME	/**crs**/app/oracle/product/11/crs	/**crs**/app/oracle/product/11/crs
RDBMS_HOME	/**oracle1**/app/oracle/product/11/db	/**oracle2**/app/oracle/product/11/db
ORACLE_SID	goodus1	goodus2
Vote disk	/dev/rdsk/c1t10d0s3 /dev/rdsk/c1t10d0s4 /dev/rdsk/c1t10d0s5	
Ocr disk	/dev/rdsk/c1t10d0s0 /dev/rdsk/c1t10d0s1	



1.3.2. Create an VIP(RAC VIP)

10g CRS 상태의 VIP 는 Oracle 에서 서비스하는 IP 이다. 즉 client 가 listener 를 통해서 접속하는 IP 임.

 # srvctl add nodeapps -n goodus1 -A goodus1-vip/255.255.255.0/hme0

 # srvctl add nodeapps -n goodus2 -A goodus2-vip/255.255.255.0/hme0

 # crsstat

HA Resource	Target	State
ora.goodus1.gsd	OFFLINE	OFFLINE
ora.goodus1.ons	OFFLINE	OFFLINE
ora.goodus1.vip	OFFLINE	OFFLINE
ora.goodus2.gsd	OFFLINE	OFFLINE
ora.goodus2.ons	OFFLINE	OFFLINE
ora.goodus2.vip	OFFLINE	OFFLINE

아래 GSD, ONS 는 H/A 구성에서 불필요 함으로 OCR 정보에서 삭제한다.

$ crs_unregister ora.goodus1.gsd

$ crs_unregister ora.goodus1.ons

$ crs_unregister ora.goodus2.gsd

$ crs_unregister ora.goodus2.ons

 # crsstat

HA Resource	Target	State
ora.goodus1.vip	OFFLINE	OFFLINE
ora.goodus2.vip	OFFLINE	OFFLINE

1.3.3. Create an action Program

 # cd $ORA_CRS_HOME/crs/public

 # ls -al

```
total 44
drwxrwxrwt    2 oracle   dba          512 Nov  4 16:35 .
drwxr-xr-x   14 root     dba          512 Nov  5 19:36 ..
-rwxrwxrwx    1 oracle   dba         1691 Nov  5 14:35 action_db1.pl
-rwxrwxrwx    1 oracle   dba         1691 Nov  5 14:35 action_db2.pl
-rwxrwxrwx    1 oracle   dba         1654 Nov  5 14:35 action_listener1.pl
-rwxrwxrwx    1 oracle   dba         1654 Nov  5 14:35 action_listener2.pl
-rwxrwxrwx    1 oracle   dba         1752 Nov  5 14:35 action_mount1.pl
```

```
-rwxrwxrwx    1 oracle     dba            1773 Nov   5 14:35 action_mount2.pl
```

=> 위 파일을 첨부된 파일로 .pl 확장자로 등록 해야 함

start, stop, check 의 script 를 정의합니다. (1.6.2 절 참조)

1.3.4. Create an application Profile

- ◆ mount (/oracle, /oradata) : 모니터링 할 file system list
- ◆ db : 모니터링 할 database
- ◆ listener : 모니터링 할 listener

clusterware 가 인식하고 ocr 정보에 등록하는 포맷형식으로 cap 파일을 생성함.

```
# crs_profile -create mount1 -t application -r ora.goodus1.vip -a
  /crs/app/oracle/product/11/crs/crs/public/action_mount1.pl -o ci=60,ra=5

# crs_profile -create db1 -t application -r mount1 -a
  /crs/app/oracle/product/11/crs/crs/public/action_db1.pl -o ci=60,ra=5

# crs_profile -create listener1 -t application -r ora.goodus1.vip -a
  /crs/app/oracle/product/11/crs/crs/public/action_listener1

# crs_profile -create mount2 -t application -r ora.goodus2.vip -a
  /crs/app/oracle/product/11/crs/crs/public/action_mount2.pl -o ci=60,ra=5

# crs_profile -create db2 -t application -r mount2 -a
  /crs/app/oracle/product/11/crs/crs/public/action_db2.pl -o ci=60,ra=5

# crs_profile -create listener2 -t application -r ora.goodus2.vip -a
  /crs/app/oracle/product/11/crs/crs/public/action_listener2
```

1.3.5. create an application profile

- action program 을 등록하면 아래 디렉토리에 cap 파일이 생성됨

```
# cd $ORA_CRS_HOME/crs/profile

# ls -al
-rw-r--r--    1 root     root          773 Nov   5 17:48 db1.cap
```

```
-rw-r--r--   1 root     root          773 Nov  5 17:50 db2.cap
-rw-r--r--   1 root     root          800 Nov  5 17:49 listener1.cap
-rw-r--r--   1 root     root          800 Nov  5 17:50 listener2.cap
-rw-r--r--   1 root     root          791 Nov  5 17:48 mount1.cap
-rw-r--r--   1 root     root          791 Nov  5 17:50 mount2.cap

# crs_register mount1
# crs_register db1
# crs_register listener1

# crs_register mount2
# crs_register db2
# crs_register listener2
```

** 등록이 완료되면 다른 노드로 **cap, pl** 파일을 **copy** 를 해야 함

($ORA_CRS_HOME/crs/profile, $ORA_CRS_HOME/crs/public)

Clusterware 의 **OCR device** 에 **crs** 가 모니터링하고 **fail** 이 발생하면 **restart** 또는 **failover** 를 수행할 **resource** 를 등록한다.

1.3.6. Set the permission on your application

◆ action program 을 수행할 user 를 oracle user 로 변경

```
# crs_setperm mount1 -u user:oracle:r-x
# crs_setperm db1 -u user:oracle:r-x
# crs_setperm listener1 -u user:oracle:r-x

# crs_setperm mount2 -u user:oracle:r-x
# crs_setperm db2 -u user:oracle:r-x
# crs_setperm listener2 -u user:oracle:r-x
```

1.3.7. Check the H/A resource

```
# crsstat
HA Resource                                   Target          State
-----------                                   ------          -----
db1                                           OFFLINE         OFFLINE
db2                                           OFFLINE         OFFLINE
```

```
listener1                                      OFFLINE      OFFLINE
listener2                                      OFFLINE      OFFLINE
mount1                                         OFFLINE      OFFLINE
mount2                                         OFFLINE      OFFLINE
ora.goodus1.vip                                OFFLINE      OFFLINE
ora.goodus2.vip                                OFFLINE      OFFLINE
```

1.4. Oracle H/A 운용

1.4.1. VIP Start

```
$ crsstat
HA Resource                                    Target       State
-----------                                    ------       -----
…
ora.goodus1.vip                                OFFLINE      OFFLINE

$ ifconfig -a
…
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 61.250.99.232 netmask ffffff00 broadcast 61.250.99.255

$ crs_start ora.goodus1.vip
Attempting to start `ora.goodus1.vip` on member `goodus1`
Start of `ora.goodus1.vip` on member `goodus1` succeeded.

$ ifconfig -a
…
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 61.250.99.232 netmask ffffff00 broadcast 61.250.99.255
hme0:1: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu 1500
index 3
        inet 61.250.99.234 netmask ffffff00 broadcast 61.250.99.255

$ crsstat
HA Resource                                    Target       State
-----------                                    ------       -----
..
ora.goodus1.vip                                ONLINE       ONLINE on goodus1
```

1.4.2. Mount Start

```
$ crsstat
HA Resource                                    Target       State
-----------                                    ------       -----
…
mount1                                         OFFLINE      OFFLINE
$ df  -k |grep ora

$ crs_start mount1
Attempting to start `mount1` on member `goodus1`
Start of `mount1` on member `goodus1` succeeded
```

```
$ df -k |grep ora
/dev/dsk/c1t1d0s0      24696312   847826 23601523       4%      /oradata1
/dev/dsk/c1t1d0s7      10327834 4526651 5697905      45%      /oracle1

$ crsstat
HA Resource                                   Target      State
-----------                                   ------      -----
db1                                           OFFLINE     OFFLINE
db2                                           OFFLINE     OFFLINE
listener1                                     OFFLINE     OFFLINE
listener2                                     OFFLINE     OFFLINE
mount1                                        ONLINE      ONLINE on goodus1
mount2                                        OFFLINE     OFFLINE
ora.goodus1.vip                               ONLINE      ONLINE on goodus1
ora.goodus2.vip                               OFFLINE     OFFLINE
```

1.4.3. Database Start

```
$ crsstat
HA Resource                                   Target      State
-----------                                   ------      -----
…
db1                                           OFFLINE     OFFLINE

$ ps  -ef |grep smon

$ crs_start db1
Attempting to start `mount1` on member `goodus1`
Start of `mount1` on member `goodus1` succeeded

$ ps -ef |grep smon
  oracle 26126       1    1 10:09:05 ?           0:00 ora_smon_goodus1
  oracle 26367 18158     0 10:09:28 pts/1       0:00 grep smon

$ crsstat
HA Resource                                   Target      State
-----------                                   ------      -----
db1                                           ONLINE      ONLINE on goodus1
db2                                           OFFLINE     OFFLINE
listener1                                     OFFLINE     OFFLINE
listener2                                     OFFLINE     OFFLINE
mount1                                        ONLINE      ONLINE on goodus1
mount2                                        OFFLINE     OFFLINE
ora.goodus1.vip                               ONLINE      ONLINE on goodus1
ora.goodus2.vip                               OFFLINE     OFFLINE
```

1.4.4. Listener Start

```
$ crsstat
HA Resource                                   Target      State
-----------                                   ------      -----
…
listener1                                     OFFLINE     OFFLINE
$ ps  -ef |grep tns
```

$ crs_start listener1
Attempting to start `listener1` on member `goodus1`
Start of `listener1` on member `goodus1` succeeded.

$ ps -ef |grep tns
oracle 27824 1 1 10:11:46 ? 0:00 /oracle1/app/oracle/product/11/db/bin/tnslsnr LISTENER1 -inherit oracle 26367 18158 0 10:09:28 pts/1 0:00 grep smon

$ crsstat

HA Resource	Target	State
db1	**ONLINE**	**ONLINE on goodus1**
db2	OFFLINE	OFFLINE
listener1	**ONLINE**	**ONLINE on goodus1**
listener2	OFFLINE	OFFLINE
mount1	**ONLINE**	**ONLINE on goodus1**
mount2	OFFLINE	OFFLINE
ora.goodus1.vip	**ONLINE**	**ONLINE on goodus1**
ora.goodus2.vip	OFFLINE	OFFLINE

1.4.5. Listener, Database, Mount, VIP STOP

$ crs_stop listener1
Attempting to stop `listener1` on member `goodus1`
Stop of `listener1` on member `goodus1` succeeded.

$ crs_stop db1
Attempting to stop `db1` on member `goodus1`
Stop of `db1` on member `goodus1` succeeded.

$ crs_stop mount1
Attempting to stop `mount1` on member `goodus1`
Stop of `mount1` on member `goodus1` succeeded.

$ crs_stop ora.goodus1.vip
Attempting to stop `ora.goodus1.vip` on member `goodus1`
Stop of `ora.goodus1.vip` on member `goodus1` succeeded.

$ crsstat

HA Resource	Target	State
db1	**OFFLINE**	**OFFLINE**
db2	**OFFLINE**	**OFFLINE**
listener1	**OFFLINE**	**OFFLINE**
listener2	**OFFLINE**	**OFFLINE**
mount1	**OFFLINE**	**OFFLINE**
mount2	**OFFLINE**	**OFFLINE**
ora.goodus1.vip	**OFFLINE**	**OFFLINE**
ora.goodus2.vip	**OFFLINE**	**OFFLINE**

1.5. H/A Fail-Over Test

1.5.1. Listener Process kill

```
$ ps -ef |grep tns
oracle 27824     1   0 10:11:46 ?           0:00 /oracle1/app/oracle/product/11/db/bin/tnslsnr
LISTENER1 -inherit

$ kill -9 27824

# while true
> do
> ps -ef |grep tns
> sleep 2
> echo "==="
> done
    root 11647  6185   0 18:45:20 pts/4     0:00 grep tns
===
    root 11666  6185   0 18:45:22 pts/4     0:00 grep tns
===
    root 11702  6185   0 18:45:24 pts/4     0:00 grep tns
===
  oracle 11724     1   1 18:45:25 ?         0:00 /oracle1/app/oracle/product/11/db/bin/tnslsnr
LISTENER1 -inherit
```

**** Check_interval 이 60 으로 설정되어 있으므로 최대 60 초 이내에 check 하여 해당 프로세스가 종료되었으면 crsd daemon 이 start 함**

1.5.2. Instance Abort

```
$ ps -ef |grep smon
oracle 26126     1   0 10:09:05 ?           0:01 ora_smon_goodus1

$ kill -9 26126

# while true
> do
> ps -ef |grep smon
> sleep 2
> echo "==="
> done
===
    root 12861  6185   0 18:47:04 pts/4     0:00 grep smon
===
    root 12930  6185   0 18:47:08 pts/4     0:00 grep smon
===
    root 12957  6185   0 18:47:10 pts/4     0:00 grep smon
  oracle 12940     1   1 18:47:08 ?         0:00 ora_smon_goodus1
```

**** Check_interval 이 60 으로 설정되어 있으므로 최대 60 초 이내에 check 하여 해당 프로세스가 종료되었으면 crsd daemon 이 start 함**

1.5.3. Oracle engine, Data Filesystem Fail

```
# fuser -ck /oradata1
/oradata1:      13616o     13041o     13002o     13000o     12944o     12942o     12940o     12938o
12936o     12934o     12913o

# umount /oradata1

# while true
> do
> df  -k |grep ora
> sleep 2
> echo "==="
> done
====
====
====
/dev/dsk/c1t1d0s0       24696312    847826 23601523        4%       /oradata1
/dev/dsk/c1t1d0s7       10327834 4526081 5698475       45%       /oracle1
```

**** Check_interval** 이 **60** 으로 설정되어 있으므로 최대 **60** 초 이내에 **check** 하여 파일시스템 마운트가 되어 있지 않으면 **crsd daemon** 마운트 시킴

**** db, listener** 까지 **start** 함

1.5.4. System Halt

$ crsstat

```
HA Resource                                   Target       State
-----------                                    ------       -----
db1                                           ONLINE       ONLINE on goodus1
db2                                           ONLINE       ONLINE on goodus2
listener1                                     ONLINE       ONLINE on goodus1
listener2                                     ONLINE       ONLINE on goodus2
mount1                                        ONLINE       ONLINE on goodus1
mount2                                        ONLINE       ONLINE on goodus2
ora.goodus1.vip                               ONLINE       ONLINE on goodus1
ora.goodus2.vip                               ONLINE       ONLINE on goodus2
```

 # halt (2 번 노드를 강제 종료 함**)**

$ crsstat

```
HA Resource                                   Target       State
-----------                                   ------       -----
db1                                           ONLINE       ONLINE on goodus1
db2                                           ONLINE       OFFLINE
```

```
listener1                                      ONLINE      ONLINE on goodus1
listener2                                      ONLINE      OFFLINE
mount1                                         ONLINE      ONLINE on goodus1
mount2                                         ONLINE      OFFLINE
ora.goodus1.vip                                ONLINE      ONLINE on goodus1
ora.goodus2.vip                                ONLINE      OFFLINE


$ crsstat

HA Resource                                    Target      State
-----------                                    ------      -----
db1                                            ONLINE      ONLINE on goodus1
db2                                            ONLINE      ONLINE on goodus1
listener1                                      ONLINE      ONLINE on goodus1
listener2                                      ONLINE      ONLINE on goodus1
mount1                                         ONLINE      ONLINE on goodus1
mount2                                         ONLINE      ONLINE on goodus1
ora.goodus1.vip                                ONLINE      ONLINE on goodus1
ora.goodus2.vip                                ONLINE      ONLINE on goodus1


$ ps -ef |grep smon
   oracle  4064     1   0 10:21:33 ?           0:01 ora_smon_goodus1
   oracle 10427     1   1 10:31:34 ?           0:01 ora_smon_goodus2
   oracle 10985 18158   0 10:32:13 pts/1       0:00 grep smon

$ ps -ef |grep tns
   oracle 27824     1   0 10:11:46 ?           0:00 /oracle1/app/oracle/product/11/db/bin/tnslsnr
LISTENER1 -inherit
   oracle 10837     1   1 10:32:02 ?           0:00 /oracle2/app/oracle/product/11/db/bin/tnslsnr
LISTENER2 -inherit
   oracle 11002 18158   0 10:32:14 pts/1       0:00 grep tns

$ df -k |grep ora
/dev/dsk/c1t1d0s0     24696312   847826 23601523     4%     /oradata1
/dev/dsk/c1t1d0s7     10327834 4526765 5697791     45%     /oracle1
/dev/dsk/c1t5d0s0     24696312   848162 23601187     4%     /oradata2
/dev/dsk/c1t5d0s7     10327834 4312445 5912111     43%     /oracle2

$ ifconfig -a
...
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 61.250.99.232 netmask ffffff00 broadcast 61.250.99.255
hme0:1: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu 1500
index 3
        inet 61.250.99.234 netmask ffffff00 broadcast 61.250.99.255
hme0:2: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu 1500
index 3
        inet 61.250.99.236 netmask ffffff00 broadcast 61.250.99.255
```

** Node 2 번을 강제 종료 되면 2 번 노드에 실행되고 있는 application resource 는 Node 1 번으로 failover 되면서 계속적으로 서비스를 수행함.

** Node 2 번이 정상적으로 부팅이 되면 수동으로 failback 을 수행해야 함.

AUTO_START=restore => 0 으로 수정해야 함

1.6. 참고 사항

1.6.1. crs_profile 명령어 Arguments (1.3.4 절 참조)

Name	Value	Description
-create	myapp	Name of the application as stored inside the OCR
-t	application	Type of OCR entry (must be Application)
-d	"Oracle Sample…"	'Long' name of the application
-r	myvip	Name of the Oracle Clusterware managed resource that must be in status ONLINE for our application to start. The VIP
-a	/tmp/myappcheck	Name of the action program used to start, stop and check the application
-o		See following entries in tables
	ci=5 ra=60	Check Interval Restart Attempts

1.6.2. Sampla Script(1.3.4 절 참조)

1) action_mount.pl

```
-- action_mount.pl
#!/usr/bin/perl
# Copyright (c) 2002, 2006, Oracle. All rights reserved.
# action_db.pl
# This perl script is the action script for start / stop / check
# the Oracle Instance in a cold failover configuration.
#
#       NAME
#       action_db.pl
#
#       DESCRIPTION
#
#       NOTES
#
# Usage:
#       rknapp 05/22/06 - Creation


# Environment settings, please modify and adapt this

$ORA_CRS_HOME    = "/crs/app/oracle/product/11/crs";
$CRS_HOME_BIN    = "/crs/app/oracle/product/11/crs/bin";
$CRS_HOME_SCRIPT = "/crs/app/oracle/product/11/crs/crs/public";
$ORACLE_HOME_BIN = "/oracle1/app/oracle/product/11/db/bin";
$ORACLE_HOME     = "/oracle1/app/oracle/product/11/db";
$ORA_SID = "goodus1";
```

```
$ORA_USER = "oracle";
$PASSWD = "tnstjd";


if ($#ARGV != 0 ) {
    print "usage: start stop check required Wn";
    exit;
}

$command = $ARGV[0];

# Database start stop check
# Start File system
if ($command eq "start" ) {
        system ("
        su - root
        tnstjd
        /usr/sbin/mount /dev/dsk/c1t1d0s0 /oradata1
        /usr/sbin/mount /dev/dsk/c1t1d0s7 /oracle1
        " );
}

# Stop File system
if ($command eq "stop" ) {
        system ("
        su - root
        tnstjd
        /usr/sbin/fuser -ck /oradata1
        /usr/sbin/fuser -ck /oracle1
        /usr/sbin/umount /oradata1
        /usr/sbin/umount /oracle1
        " );
}

# Check filesystem
if ($command eq "check" ) {
        check();
}

sub check {
my($check_proc,$process) = @_;
$mount_ora = "/oracle1";
$mount_oradata = "/oradata1";
$check_ora = qx(df -k |grep oracle1| awk '{print W$6}');
$check_oradata = qx(df -k |grep oradata1| awk '{print W$6}');
        chomp($check_ora);
        chomp($check_oradata);
        if (($mount_ora eq $check_ora) && ($mount_oradata eq $check_oradata)) {
                exit 0;
        } else {
                exit 1;
        }
}
```

2) action_listener.pl

```
#!/usr/bin/perl
#!/usr/bin/perl
# Copyright (c) 2002, 2006, Oracle. All rights reserved.
# action_listener.pl
# This perl script is the action script for start / stop / check
# the Oracle Listener in a cold failover configuration.
#
#       NAME
#       action_listener.pl
#
#       DESCRIPTION
#
#       NOTES
#
# Usage:
#       rknapp 05/22/06 - Creation


# Environment settings, please modify and adapt this

$ORA_CRS_HOME    = "/crs/app/oracle/product/11/crs";
$CRS_HOME_BIN    = "/crs/app/oracle/product/11/crs/bin";
$CRS_HOME_SCRIPT = "/crs/app/oracle/product/11/crs/crs/public";
$ORACLE_HOME_BIN = "/oracle1/app/oracle/product/11/db/bin";
$ORACLE_HOME     = "/oracle1/app/oracle/product/11/db";
$ORA_SID = "goodus1";
$ORA_USER = "oracle";

if ($#ARGV != 0 ) {
    print "usage: start stop check required Wn";
    exit;
}

$command = $ARGV[0];

# Listener start / stop check
# start listener
if ($command eq "start") {
        system ("
        su - $ORA_USER << EOF
        export ORACLE_HOME=$ORACLE_HOME
        $ORACLE_HOME_BIN/lsnrctl start LISTENER1
        EOF");
}

# stop listener
if ($command eq "stop") {
        system ("
        su - $ORA_USER << EOF
        export ORACLE_HOME=$ORACLE_HOME
        $ORACLE_HOME_BIN/lsnrctl stop LISTENER1
```

```
        EOF");

}

# check listener
if ($command eq "check") {
        check_listener();
}

sub check_listener {
my($check_proc_listener,$process_listener) = @_;
$process_listener = "$ORACLE_HOME_BIN/tnslsnr LISTENER1";

$check_proc_listener = qx(ps -aef | grep "tnslsnr LISTENER1" | grep -v grep | awk
    '{print W$8,W$9}');
        chomp($check_proc_listener);
        if ($process_listener eq $check_proc_listener) {
                exit 0;
        } else {
                exit 1;
        }
}
```

3) action_db1.pl

```
#!/usr/bin/perl
# Copyright (c) 2002, 2006, Oracle. All rights reserved.
# action_db.pl
# This perl script is the action script for start / stop / check
# the Oracle Instance in a cold failover configuration.
#
#       NAME
#       action_db.pl
#
#       DESCRIPTION
#
#       NOTES
#
# Usage:
#       rknapp 05/22/06 - Creation


# Environment settings, please modify and adapt this

$ORA_CRS_HOME    = "/crs/app/oracle/product/11/crs";
$CRS_HOME_BIN    = "/crs/app/oracle/product/11/crs/bin";
$CRS_HOME_SCRIPT = "/crs/app/oracle/product/11/crs/crs/public";
$ORACLE_HOME_BIN = "/oracle1/app/oracle/product/11/db/bin";
$ORACLE_HOME     = "/oracle1/app/oracle/product/11/db";
$ORA_SID = "goodus1";
$ORA_USER = "oracle";


if ($#ARGV != 0 ) {
```

```
    print "usage: start stop check required ₩n";
    exit;
}

$command = $ARGV[0];
 # Database start stop check
# Start database
if ($command eq "start" ) {
        system ("
        su - $ORA_USER << EOF
export ORACLE_SID=$ORA_SID
        export ORACLE_HOME=$ORACLE_HOME
$ORACLE_HOME/bin/sqlplus /nolog
        connect / as sysdba
        startup
        quit
        EOF" );
}

# Stop database
if ($command eq "stop" ) {
        system ("
        su - $ORA_USER << EOF
        export ORACLE_SID=$ORA_SID
        export ORACLE_HOME=$ORACLE_HOME
        $ORACLE_HOME/bin/sqlplus /nolog
        connect / as sysdba
        shutdown immediate
        quit
        EOF" );
}

# Check database
if ($command eq "check" ) {
        check();
}

sub check {
my($check_proc,$process) = @_;
$process = "ora_pmon_$ORA_SID";
$check_proc = qx(ps -aef | grep ora_pmon_$ORA_SID | grep -v grep | awk '{print ₩$8}');
        chomp($check_proc);
        if ($process eq $check_proc) {
                exit 0;
        } else {
                exit 1;
        }
}
```

4) crs_stat script

```
#!/usr/bin/perl
#!/usr/bin/ksh
#
# Sample 10g CRS resource status query script
#
# Description:
#    - Returns formatted version of crs_stat -t, in tabular
#      format, with the complete rsc names and filtering keywords
#   - The argument, $RSC_KEY, is optional and if passed to the script, will
#     limit the output to HA resources whose names match $RSC_KEY.
# Requirements:
#   - $ORA_CRS_HOME should be set in your environment

RSC_KEY=$1
QSTAT=-u
AWK=/usr/xpg4/bin/awk    # if not available use /usr/bin/awk

# Table header:echo ""
$AWK W
  'BEGIN {printf "%-45s %-10s %-18sWn", "HA Resource", "Target", "State";
          printf "%-45s %-10s %-18sWn", "-----------", "------", "-----";}'

# Table body:
$ORA_CRS_HOME/bin/crs_stat $QSTAT | $AWK W
 'BEGIN { FS="="; state = 0; }
  $1~/NAME/ && $2~/'$RSC_KEY'/ {appname = $2; state=1};
  state == 0 {next;}
  $1~/TARGET/ && state == 1 {apptarget = $2; state=2;}
  $1~/STATE/ && state == 2 {appstate = $2; state=3;}
  state == 3 {printf "%-45s %-10s %-18sWn", appname, apptarget, appstate; state=0;}'
```