10g R2 Clusterware 를 이용한 3rd Application Fail-Over

일시 : 2006. 10.

Contents

I. Introduction

II. The HA Framework

III. Fail-Over Sample Test

IV. WebServer Fail-Over Test

V. 기타내용 및 QnA

I. Introduction

Oracle Clusterware ...

a. kill -9 'LGWR Process' ➔ Restart Automatically !
kill -9 'listerner PID' ➔ Restart Automatically !

b. kill 'oracle.exe' by Windows Task Manager ➔ Restart Automatically !
kill 'tnslsnr.exe' by Windows Task Manager ➔ Restart Automatically !

c. When a Node dies ➔ VIP fails over to a different node !!

10g Release 2 is Protect 3rd Party Applications !!

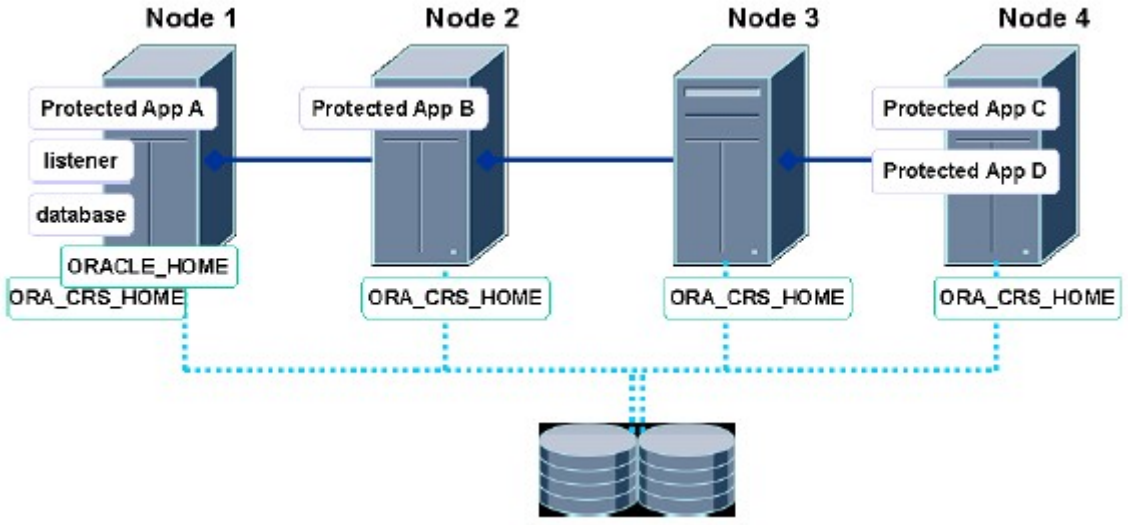
** Oracle Clusterware 를 사용하여 Single Instance 환경의 HA 기능도 제공하며 Single Node / Local FileSystem 을 ASM 또는 Cluster FileSystem 인 OCFS등으로 전환하여 High Availability 기능을 이용할 수 있다.

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

II. The HA Framework

a. Shared Resources
- A common disks : OCR & Vote for Oracle Clusterwaere system files
- A common network interconnect
- Same OS

b. 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

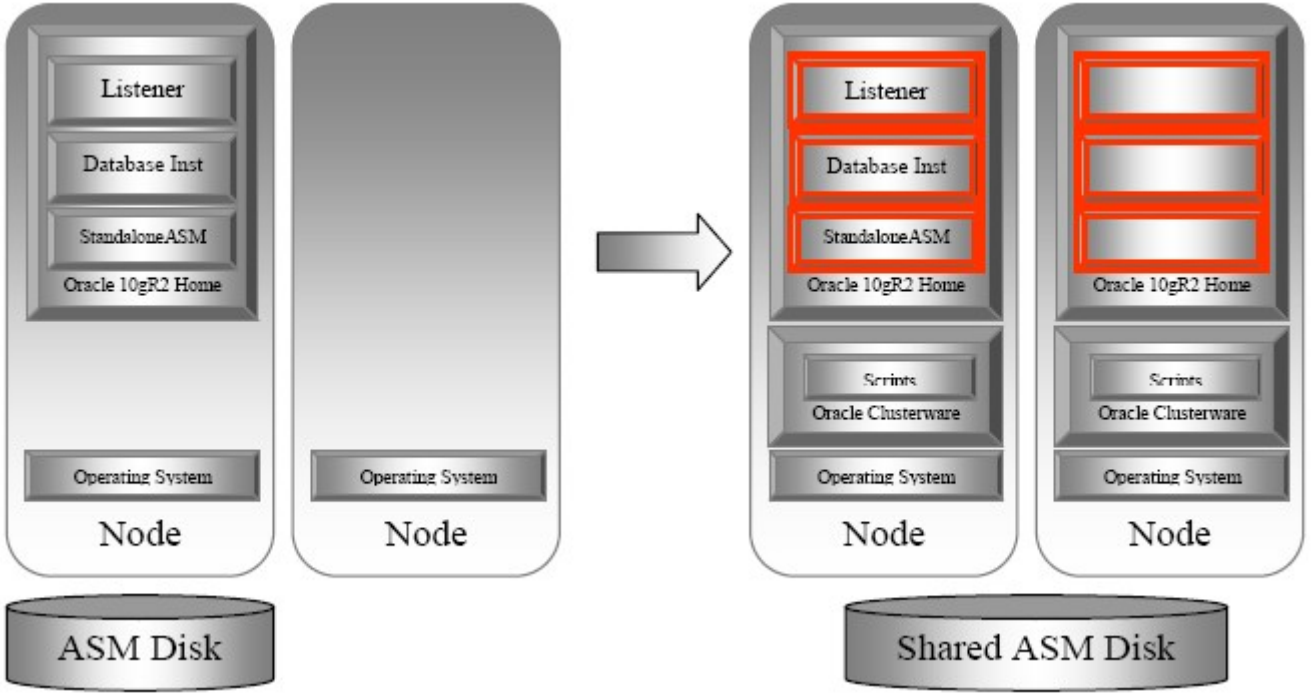


II. The HA Framework

a. Install Oracle S/W Only into a new Oracle Clusterware HOME on the 2nd node

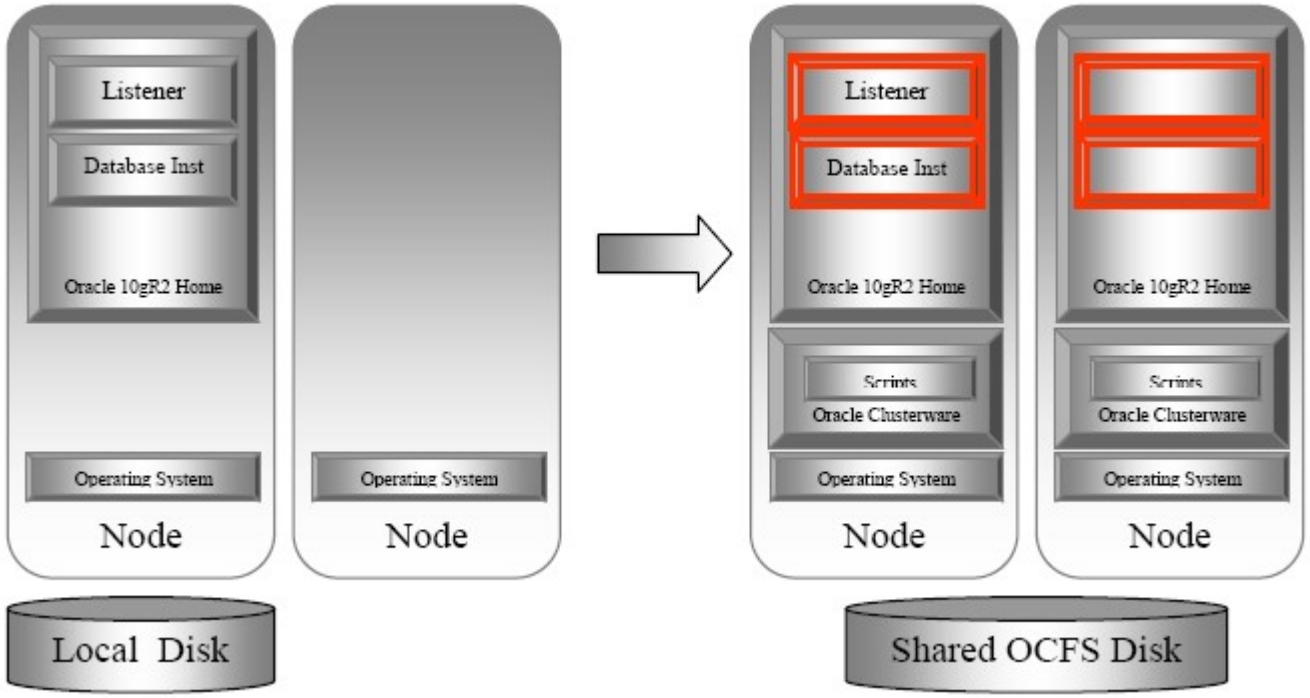
b. Modify Oracle Configurations ...
- orapwd
- initASM.ora
- init<SID>.ora
- listener.ora , tnsnames.ora

c. Create / Register / Start / Relocate Resources
- $ORA_CRS_HOME/crs/public



II. The HA Framework

a. Install Oracle S/W Only into a new Oracle Clusterware HOME on the 2nd node
b. Install Cluster Filesystem both nodes
c. Modify Oracle Configurations ...
   - orapwd
   - initASM.ora
   - init<SID>.ora
   - listener.ora , tnsnames.ora
d. Create / Register / Start / Relocate Resources
   - $ORA_CRS_HOME/crs/public



II. The HA Framework

Using the Application Framework

1. Create an application VIP
   required if the application is access via network clients

2. Create an action Program
   Oracle Clusterware to start, stop & query the status
   C, Java or 기타 scripting language

3. Create an application Profile
   Describes the process and limits to protect

4. Register the application
   Register the Application Profile with Oracle Clusterware

III. Fail-Over Sample Test

Create an Application VIP

1. Create an profile for VIP

```
$ORA_CRS_HOME/bin/crs_profile –create myvip ₩
–t application –a $ORA_CRS_HOME/bin/usrvip ₩
–o oi=eth0,ov=61.250.99.230,on=255.255.255.0
```

➔ $ORA_CRS_HOME/crs/public/myvip.cap 파일이 자동생성됨

2. Register the VIP with Oracle Clusterware

```
$ORA_CRS_HOME/bin/crs_register myvip
```

3. Change the owner of resource as a root

```
$ORA_CRS_HOME/bin/crs_setperm myvip –o root
```

4. Allow to execute the script

```
$ORA_CRS_HOME/bin/crs_setperm myvip –u user:oracle10:r-x
```

5. Start VIP

```
$ORA_CRS_HOME/bin/crs_start myvip
```

```
[oracle10@rac1 tmp]$ /sbin/ifconfig –a
  eth0:2    Link encap:Ethernet  HWaddr 00:04:76:6F:3E:FE
            inet addr:61.250.99.230  Bcast:61.250.99.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            Interrupt:10 Base address:0xb800
```

III. Fail-Over Sample Test

Create an Action Program

1. Create an Application Profile

```
$ORA_CRS_HOME/bin/crs_profile –create myapp ₩
–t application –d “Oracle Sample Test Action Program” ₩
–r myvip –a  /tmp/myappcheck –o ci=60,ra=5
```

➔ $ORA_CRS_HOME/crs/public/myapp.cap 파일이 자동생성됨
➔ /tmp/myappcheck 는 실행 Binary 파일형태로 RAC 모든 Node 에 있어야 함.!!.

2. Register the VIP with Oracle Clusterware

```
$ORA_CRS_HOME/bin/crs_register myapp
```

3. Change the profile stored in OCR dynamically

```
$ORA_CRS_HOME/bin/crs_register myapp –update –o ra=3
```

4. Querying the state of the application

```
$ORA_CRS_HOME/bin/crs_stat myapp
```

5. Start application

```
$ORA_CRS_HOME/bin/crs_start myapp
[oracle10@rac1 tmp]$ crs_stat –t –v
Name           Type           R/RA   F/FT   Target    State     Host
----------------------------------------------------------
myapp          application    0/3    0/0    ONLINE    ONLINE    rac1
myvip          application    0/1    0/0    ONLINE    ONLINE    rac1
```

III. Fail-Over Sample Test

Relocate the application

1. Relocate the application

$ORA_CRS_HOME/bin/crs_relocate –f myapp

➔ 현재 서비스중인 Resource 를 수동으로 다른 Node 로 Fail-Over 시킴.

➔ 해당 Resource 의 실행 Binary 파일형태로 RAC 모든 Node 에 있어야 F/O 됨.

```
[oracle10@rac2 tmp]$ crs_relocate -f myapp
Attempting to stop `myapp` on member `rac1`
Stop of `myapp` on member `rac1` succeeded.
Attempting to stop `myvip` on member `rac1`
Stop of `myvip` on member `rac1` succeeded.
Attempting to start `myvip` on member `rac2`
Start of `myvip` on member `rac2` succeeded.
Attempting to start `myapp` on member `rac2`
Start of `myapp` on member `rac2` succeeded.
[oracle10@rac2 tmp]$ crs_stat -t -v
Name           Type           R/RA   F/FT   Target    State     Host
----------------------------------------------------------
myapp          application    0/3    0/0    ONLINE    ONLINE    rac2
myvip          application    0/1    0/0    ONLINE    ONLINE    rac2
```

2. Status / Stop / Remove Resource

$ORA_CRS_HOME/bin/crs_stat -v myapp

$ORA_CRS_HOME/bin/crs_stop myapp

$ORA_CRS_HOME/bin/crs_unregister myapp ← cap 파일존재, Recreate 는 불필요!!

III. Fail-Over Sample Test

Testing Application on Failure

1. Register & Start Application on Failure

```
$ORA_CRS_HOME/bin/crs_register myapp
$ORA_CRS_HOME/bin/crs_start myapp
```

2. Testing Application Failure

```
$ telnet 61.250.99.230 8087
$ ps –aef | grep myapp
 oracle10      1392   3076  0  13;17  pts/4  00 : 00 : 00   /tmp/myapp
$ kill –9 1392
$ telnet 61.250.99.230 8087
```

← The Action program only checks the state of the protected service every 5 seconds
Oracle Clusterware detected the application failure and restarted it.

III. Fail-Over Sample Test

Testing Node Failure

1. Check the Application

$ORA_CRS_HOME/bin/crs_stat myapp
NAME=myapp
TYPE=application
TARGET=ONLINE
STATE=ONLINE on rac1

2. Testing Application Failure

$ telnet 61.250.99.230 8087

== Power-Off the rac1 node ==

$ telnet 61.250.99.230 8087
$ORA_CRS_HOME/bin/crs_stat myapp
NAME=myapp
TYPE=application
TARGET=ONLINE
STATE=ONLINE on rac2

← The Action program only checks the state of the protected service every 5 seconds
Oracle Clusterware detected the application failure and restarted it.

IV. WebServer Fail-Over Test

Install Apache WebServer 2 Node

1. Download Apache binary
   www.apache-kr.org ➔ download ➔ apache_1.3.33.tar.gz

2. Compile Apache WebServer
   tar xvzf apache_1.3.33.tar.gz

   ./configure –prefix=/fw1/SRC/httpd ₩
   --enable-rules=SHARED_CORE ₩
   --enable-module=so

   ./make
   ./make install

3. Modify httpd.conf
   ServerName ➔ 61.250.99.230
   Port ➔ 8080

IV. WebServer Fail-Over Test

Testing Apache WebServer Resource

1. Create / Start VIP Resource ← 8 Page 참조

2. Create / Register / Start Apache Resource

```
$ORA_CRS_HOME/bin/crs_profile –create apache –t application –r myvip ₩
  –a /fw1/SRC/httpd/bin/apache_start –o ci=60,ra=5
$ORA_CRS_HOME/bin/crs_register apache
$ORA_CRS_HOME/bin/crs_setperm apache –u user:oracle10:r-x
```

← 만약 Apache Port 를 1024 이하라면 Resource owner 를 root 로 바꾸어야 한다.

```
% more apache_start
 #!/bin/sh
  /fw1/SRC/httpd/bin/apachectl start
```

3. Start Apache Resource

```
$ORA_CRS_HOME/bin/crs_start apache
[oracle10@rac1 bin]$ crs_stat –t –v
Name          Type          R/RA   F/FT   Target    State     Host
----------------------------------------------------------------
apache        application   0/5    0/0    ONLINE    ONLINE    rac1
myvip         application   0/1    0/0    ONLINE    ONLINE    rac1
[oracle10@rac1 bin]$ ps –ef |grep httpd
oracle10 13016     1  0 19:41 ?       00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10 13025 13016  0 19:41 ?       00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10 13026 13016  0 19:41 ?       00:00:00 /fw1/SRC/httpd/bin/httpd
…
```

← http://61.250.99.230:8080 으로 접속 확인 !!

IV. WebServer Fail-Over Test

Testing Apache WebServer Resource

4. Relocate Apache Resource

```
[oracle10@rac1 bin]$ crs_relocate -f apache
Attempting to stop `apache` on member `rac1`
Stop of `apache` on member `rac1` succeeded.
Attempting to stop `myvip` on member `rac1`
Stop of `myvip` on member `rac1` succeeded.
Attempting to start `myvip` on member `rac2`
Start of `myvip` on member `rac2` succeeded.
Attempting to start `apache` on member `rac2`
Start of `apache` on member `rac2` succeeded.

[oracle10@rac2 bin]$ crs_stat -t -v
Name           Type           R/RA   F/FT   Target    State     Host
----------------------------------------------------------------------
apache         application    0/5    0/0    ONLINE    ONLINE    rac2
myvip          application    0/1    0/0    ONLINE    ONLINE    rac2

[oracle10@rac2 bin]$ ps -ef |grep httpd
oracle10  5717     1  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5718  5717  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5719  5717  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
...
```

← http://61.250.99.230:8080 으로 접속 확인 !!

IV. WebServer Fail-Over Test

Testing Apache WebServer Resource

5. Apache Process Kill

```
[oracle10@rac2 bin]$ ps -ef |grep httpd
oracle10  5717     1  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5718  5717  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5719  5717  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5720  5717  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5721  5717  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5722  5717  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  8479  5717  0 15:09 ?        00:00:00 /fw1/SRC/httpd/bin/httpd

[oracle10@rac2 bin]$ kill -9 5717     ← Kill 시키면 바로 Restart 가 됨.

[oracle10@rac2 bin]$ ps -ef |grep httpd
oracle10  5718     1  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5719     1  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5720     1  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5721     1  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  5722     1  0 15:07 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
oracle10  8479     1  0 15:09 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
```

IV. WebServer Fail-Over Test

Testing Apache WebServer Resource

6. Node Down Test
== Power-Off the rac2 node ==

```
[root@rac2 ~]# sync
[root@rac2 ~]# shutdown -r now
Broadcast message from root (pts/0) (Tue Oct 10 15:15:16 2006):
The system is going down for reboot NOW!
```

← 조금 후에...

```
[oracle10@rac1 bin]$ crs_stat -t -v
  Name           Type           R/RA   F/FT   Target    State     Host
  ----------------------------------------------------------------------
  apache         application    0/5    0/0    ONLINE    ONLINE    rac1
  myvip          application    0/1    0/0    ONLINE    ONLINE    rac1
[oracle10@rac1 bin]$ ps -ef |grep httpd
  oracle10 30531     1  0 19:53 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
  oracle10 30532 30531  0 19:53 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
  oracle10 30533 30531  0 19:53 ?        00:00:00 /fw1/SRC/httpd/bin/httpd
[oracle10@rac1 bin]$ /sbin/ifconfig -a
  eth0:2    Link encap:Ethernet  HWaddr 00:04:76:6F:3E:FE
            inet addr:61.250.123.12  Bcast:61.250.123.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            Interrupt:10 Base address:0xb800

  eth0:3    Link encap:Ethernet  HWaddr 00:04:76:6F:3E:FE
            inet addr:61.250.99.230  Bcast:61.250.99.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            Interrupt:10 Base address:0xb800
```

V. 기타내용 및 QnA

Profile Arguments ... ← 9 페이지 Sample

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

V. 기타내용 및 QnA

```
-- myapp.c                ← gcc myapp.c –o myapp
#include <sys/socket.h>
#include <netinet/in.h>
#include <time.h>
#define PORT 8087

int main() {
int insock, outsock, addrlen;
struct sockaddr_in inaddr, from;
time_t timeval;
char buff[100];
int nallowreuse = 1;

insock = socket( PF_INET, SOCK_STREAM, 0);
inaddr.sin_family = AF_INET;
inaddr.sin_addr.s_addr = htonl(INADDR_ANY);
inaddr.sin_port = htons(PORT);
setsockopt(insock, SOL_SOCKET, SO_REUSEADDR, (char *)&nallowreuse, sizeof(nallowreuse));
bind(insock, (struct sockaddr *) &inaddr, sizeof(inaddr));
listen(insock,5);

while (strncmp(buff, "EXIT", 4) != 0) {
addrlen = sizeof(inaddr);
outsock = accept( insock, (struct sockaddr*) &from, &addrlen);
read(outsock, buff, sizeof(buff));
if (strncmp(buff, "EXIT",4) == 0)
close(insock);
else
{
time(&timeval);
strcpy(buff, ctime(&timeval));
write(outsock,buff,strlen(buff));
}
close(outsock);
}
}
```

V. 기타내용 및 QnA

-- myappcheck.c ← gcc myappcheck.c –o myappcheck

```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define SUCCESS 0
#define FAILURE 1
#define PORT 8087
#define MYPROCESS "//tmp//myapp &"

int main(int argc, char ** argv[])
{
char command[128];
int ret;
int sockfd;
int len;
struct sockaddr_in address;

strcpy(command, (char *)argv[1]);
ret = FAILURE;
if (strcasecmp(command, "start") == 0)
{
system(MYPROCESS);
ret = SUCCESS;
}
else if (strcasecmp(command, "stop") == 0)
{
sockfd = socket(PF_INET, SOCK_STREAM, 0);
address.sin_family = AF_INET;
address.sin_addr.s_addr = inet_addr("127.0.0.1");
address.sin_port = htons(PORT);
if (connect(sockfd, (struct sockaddr *)&address, sizeof(address)) != -
1)
{
 write(sockfd, "EXIT", 4);
sleep(2);
close(sockfd);
}
ret = SUCCESS;
}
else if (strcasecmp(command, "check") == 0)
{
sockfd = socket(PF_INET, SOCK_STREAM, 0);
address.sin_family = AF_INET;
address.sin_addr.s_addr = inet_addr("127.0.0.1");
address.sin_port = htons(PORT);
if (connect(sockfd, (struct sockaddr *)&address, sizeof(address)) != -
1)
ret = SUCCESS;
}
return(ret);
}
```

V. 기타내용 및 QnA

-- $ORA_CRS_HOME/crs/public/myapp.cap 파일 내용

```
NAME=myapp
TYPE=application
ACTION_SCRIPT=/tmp/myappcheck
ACTIVE_PLACEMENT=0
AUTO_START=restore
CHECK_INTERVAL=5
DESCRIPTION=Oracle Date Time Service
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
OPTIONAL_RESOURCES=
PLACEMENT=balanced
REQUIRED_RESOURCES=myvip
RESTART_ATTEMPTS=60
SCRIPT_TIMEOUT=60
START_TIMEOUT=0
STOP_TIMEOUT=0
UPTIME_THRESHOLD=7d
USR_ORA_ALERT_NAME=
USR_ORA_CHECK_TIMEOUT=0
USR_ORA_CONNECT_STR=/ as sysdba
USR_ORA_DEBUG=0
USR_ORA_DISCONNECT=false
USR_ORA_FLAGS=
USR_ORA_IF=
USR_ORA_INST_NOT_SHUTDOWN=
USR_ORA_LANG=
USR_ORA_NETMASK=
USR_ORA_OPEN_MODE=
USR_ORA_OPI=false
USR_ORA_PFILE=
USR_ORA_PRECONNECT=none
USR_ORA_SRV=
USR_ORA_START_TIMEOUT=0
USR_ORA_STOP_MODE=immediate
USR_ORA_STOP_TIMEOUT=0
USR_ORA_VIP=
```

V. 기타내용 및 QnA

```
-- action_db.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 = "/scratch/oracle-10.2.0/crs";
$CRS_HOME_BIN = "/scratch/oracle-10.2.0/crs/bin";
$CRS_HOME_SCRIPT = "/scratch/oracle-10.2.0/crs/crs/public";
$ORACLE_HOME_BIN = "/scratch/oracle-10.2.0/db/bin";
$ORACLE_HOME = "/scratch/oracle-10.2.0/db";
$ORA_SID = "orcl";
$ORA_ASM_SID = "+ASM";
$ORA_USER = "oracle10";
if ($#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
$ORACLE_HOME_BIN/sqlplus /nolog
connect / as sysdba
startup
quit
EOF" );
}
# Stop database
if ($command eq "stop" ) {
system ("
su - $ORA_USER << EOF
export ORACLE_SID=$ORA_SID
$ORACLE_HOME_BIN/sqlplus /nolog
connect / as sysdba
shutdown immediate
quit
EOF" );
}
# Check database
if ($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;
}
}
```

V. 기타내용 및 QnA

-- action_asm.pl

```
#!/usr/bin/perl
# Copyright (c) 2002, 2006, Oracle. All rights reserved.
# action_asm.pl
# This perl script is the action script for start / stop / check
# the Oracle ASM Instance in a cold failover configuration.
#
# NAME
# action_asm.pl
#
# DESCRIPTION
#
# NOTES
#
# Usage:
# rknapp 05/22/06 - Creation
# Environment settings, please modify and adapt this
$ORA_CRS_HOME = "/scratch/oracle-10.2.0/crs";
$CRS_HOME_BIN = "/scratch/oracle-10.2.0/crs/bin";
$CRS_HOME_SCRIPT = "/scratch/oracle-10.2.0/crs/crs/public";
$ORACLE_HOME_BIN = "/scratch/oracle-10.2.0/db/bin";
$ORACLE_HOME = "/scratch/oracle-10.2.0/db";
$ORA_SID = "orcl";
$ORA_ASM_SID = "+ASM";
$ORA_USER = "oracle10";
if ($#ARGV != 0 ) {
print "usage: start stop check or start_asm stop_asm check_asm
required
₩n";
exit;
}
$command = $ARGV[0];
# ASM database start stop check
# Start ASM database
if ($command eq "start" ) {
system ("
su - $ORA_USER << EOF
export ORACLE_SID=$ORA_ASM_SID
$ORACLE_HOME_BIN/sqlplus /nolog
connect / as sysdba
startup
quit
EOF" );
}
# Stop ASM database
if ($command eq "stop" ) {
system ("
su - $ORA_USER << EOF
export ORACLE_SID=$ORA_ASM_SID
$ORACLE_HOME_BIN/sqlplus /nolog
connect / as sysdba
shutdown immediate
quit
EOF" );
}
# Check ASM database
if ($command eq "check") {
check_asm();
}
sub check_asm {
my($check_proc_asm,$process_asm) = @_;
$process_asm = "asm_pmon_$ORA_ASM_SID";
$check_proc_asm = qx(ps -aef | grep asm_pmon_$ORA_ASM_SID |
grep -v grep | awk
'{print ₩$8}');
chomp($check_proc_asm);
if ($process_asm eq $check_proc_asm) {
exit 0;
} else {
exit 1;
}
}
```

V. 기타내용 및 QnA

-- action_listener.pl

```
#!/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 = "/scratch/oracle-10.2.0/crs";
$CRS_HOME_BIN = "/scratch/oracle-10.2.0/crs/bin";
$CRS_HOME_SCRIPT = "/scratch/oracle-10.2.0/crs/crs/public";
$ORACLE_HOME_BIN = "/scratch/oracle-10.2.0/db/bin";
$ORACLE_HOME = "/scratch/oracle-10.2.0/db";
$ORA_SID = "orcl";
$ORA_ASM_SID = "+ASM";
$ORA_USER = "oracle10";
if ($#ARGV != 0 ) {
print "usage: start stop check required ₩n";
exit;
}
$command = $ARGV[0];

# Listener start / stop check
# start listener
if ($command eq "start") {
system ("
su - $ORA_USER << EOF
$ORACLE_HOME_BIN/lsnrctl start
EOF");
}
# stop listener
if ($command eq "stop") {
system ("
su - $ORA_USER << EOF
$ORACLE_HOME_BIN/lsnrctl stop
EOF");
}
# check listener
if ($command eq "check") {
check_listener();
}
sub check_listener {
my($check_proc_listener,$process_listener) = @_;
$process_listener = "$ORACLE_HOME_BIN/tnslsnr LISTENER";
$check_proc_listener = qx(ps -aef | grep "tnslsnr LISTENER" | grep -
v grep |
awk '{print ₩$8,₩$9}');
chomp($check_proc_listener);
if ($process_listener eq $check_proc_listener) {
exit 0;
} else {
exit 1;
}
}
```

V. 기타내용 및 QnA

QnA

감 사 합 니 다