Oracle 11gR2 RAC to RAC Active DataGuard (with ASM+Rawdevice)

Author	**강경구**
Creation Date	**2010-08-05**
Last Updated	
Version	**1.0**

Version	변경일자	변경자(작성자)	주요내용
1	2010-08-05	강경구	문서 최초 작성

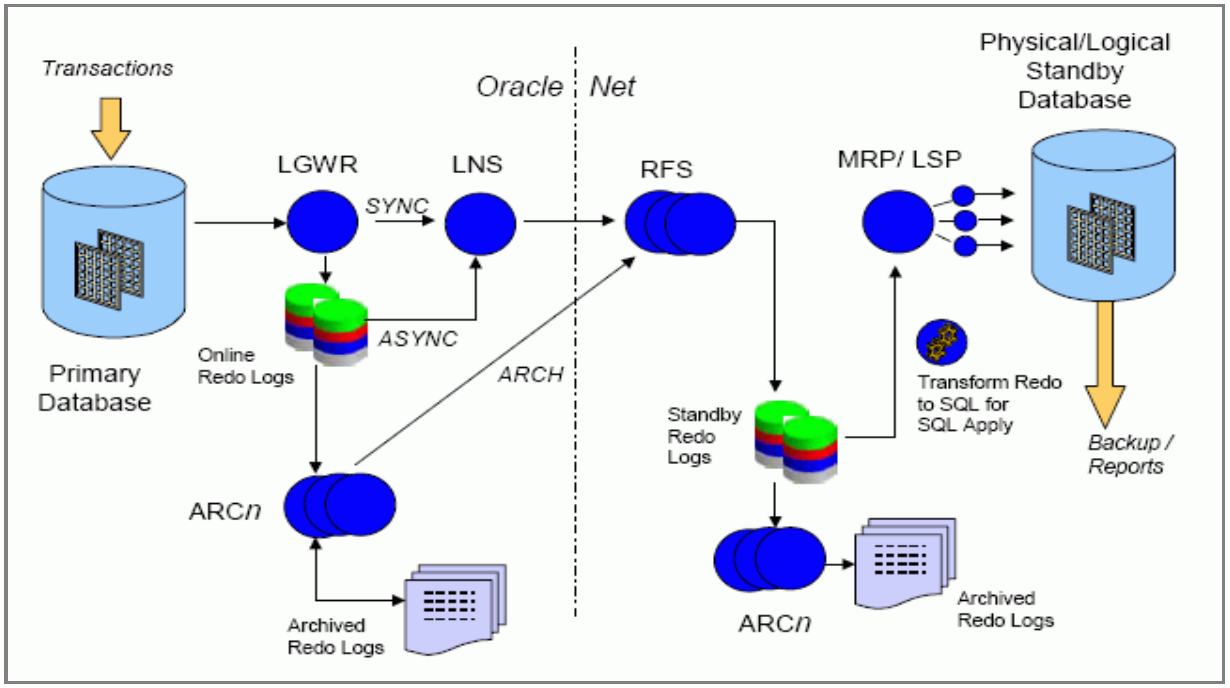
Goodus
Breathing Life into Technology

Contents

1. Oracle 11g DataGuard

이번 호에서는 데이터 보호와 더불어 Standby 시스템이 Standby 역할과 동시에 Production 과 QA 기능을 지원하는 통합 HA/DR 솔루션을 제공함으로써 기존 재해복구 패러다임의 근본적인 변화를 가져온 **Data Guard 11g 의 신기능과 Physical Active Dataguard 구축방법을** 알아보겠습니다. 일반적인 DataGuard 에 대한 소개는 '**굿어스 기술노트 21 호 Data Guard**'를 참조해 주십시요.

1.1. 11g Data Guard 구성



[11g Dataguard Architecture]

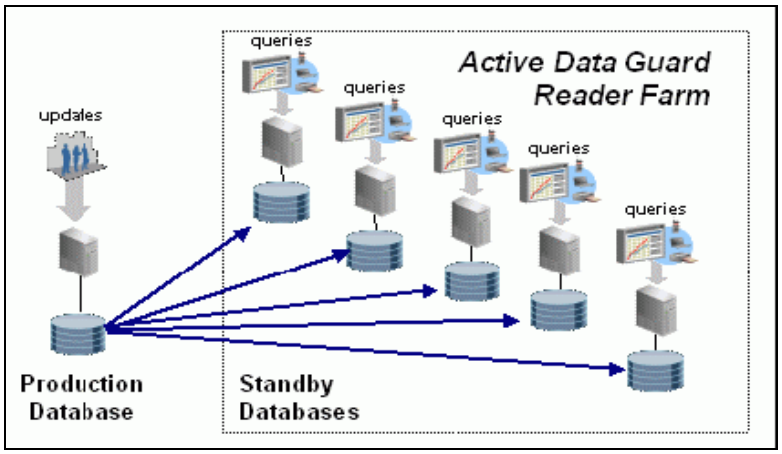
위의 그림을 보시는 것 과 같이 11g Data Guard 는 오라클 데이터베이스 프로세스를 이용하여 고가용성과 재해복구에 필요한 자동화를 실현합니다.

Primary Database 에서 Data Guard 는 LogWriter Network Server (LNS)라는 특수 Background 프로세스를 사용해 Log Writer 가 작성하는 redo 데이터를 캡처해 동기 또는 비동기로 데이터를 스탠바이 데이터베이스에 전송합니다. LNS 프로세스는 Log Writer 를 전송 오버헤드와 네트워크 방해로부터 분리합니다. 동기 redo 전송 (SYNC)를 위해서는 Primary Database 에 있는 Log Writer 가 LNS 로부터 스탠바이가 redo 데이터를 받았으며 기다렸다가 클라이언트 애플리케이션에 실행을 확인하기 전에 스탠바이 redo 로그에 데이터를 작성했다는 확인을 기다려야 합니다. 이렇게 함으로써 모든 실행 트랜잭션이 디스크에서 이루어지며 스탠바이 위치에서 보호됩니다. 비동기 redo 전송 (ASYNC)를 위해서 기본에 있는 Log Writer 가 스탠바이 데이터모드로부터 redo 가 디스크에 작성되었으며 redo 전송과 비동기인 클라이언트 애플리케이션에 실행이 확인되었다는 확인을 기다릴 필요가 없습니다. 뿐만 아니라 ASYNC 는 매우 효율적으로 redo 를 스탠바이 데이터베이스에 스트림해 네트워크 확인으로 인해 야기되는 네트워크 작업량 감소의 원인이 되는 오버헤드를 제거합니다.

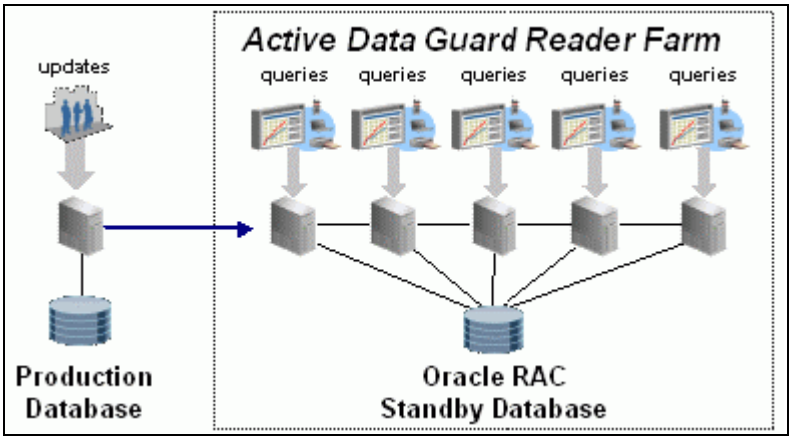
Primary Database 와 Standby Database 가 분리되는 경우 (네트워크 장애나 스탠바이 서버 장애)와 선택된 보호 모드에 따라서 Primary Database 가 트랜잭션을 계속해서 프로세스하고 새로운 네트워크 연결이 이루어질 때까지 스탠바이에 발송될 수 없는 redo 데이터 백로그를 축적합니다 (아카이브 로그 갭이라고 함). 이러한 상태에서는 Data Guard 가 계속해서 스탠바이 데이터베이스 상태를 모니터링하고, 연결이 다시 이루어지는 때를 감지하고, Primary Database 를 사용해 스탠바이 데이터베이스를 자동을 재동기화 해서 구성을 가능한 신속하게 보호 상태로 복구합니다. 갭을 해결하는데 필요한 로그 파일을 발송하는데 아카이브 (ARCn) 프로세스를 항상 사용합니다.

1.2. Oracle 11g Data Guard 와 Oracle RAC

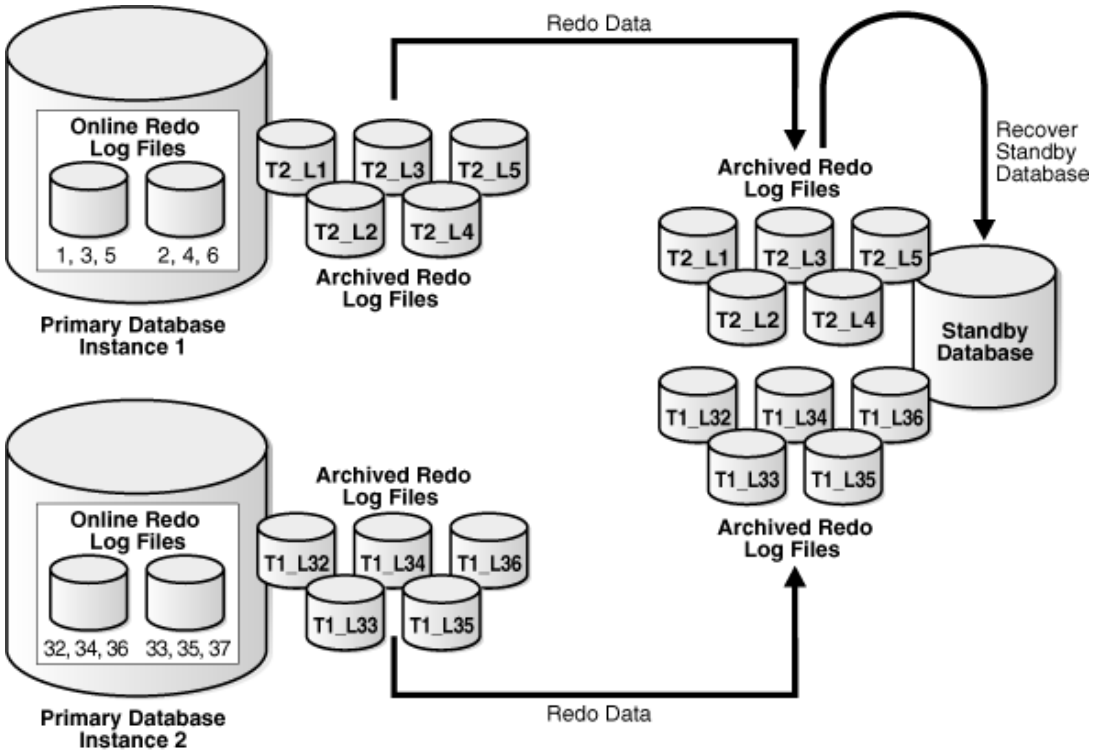
Data Guard 와 Oracle RAC 는 가장 높은 수준의 확장성, 가용성, 데이터 보호를 제공하는 상호보완적인 기술입니다. Cascaded Destinations 사용에 적용되는 제약을 제외하고 (위에서 설명) Oracle RAC 와 Data Guard 사이의 통합은 원활하게 이루어집니다. Oracle RAC 와 단일 노드 데이터베이스를 결합해 Data Guard 구성 참여하고 역할을 맡을 수 있습니다. Oracle RAC 는 업계 유일의 작업부하 관리와 확장성 기능을 제공하는 동시에 서버 장애에 대비해 보호하는 이상적인 HA 솔루션을 제공합니다. Data Guard 는 완전한 스토리지 어레이 장애, 운영자 실수, Oracle RAC 노드에 걸쳐 롤링 방식으로 실시될 수 없는 예상된 유지, 사이트 장애를 가져오는 여러 개의 연관된 장애로 인한 다운타임을 최소화하는 완전한 중복성 (redundancy)을 갖추고 데이터 가용성과 보호의 수순을 다시 한 차원 높입니다.



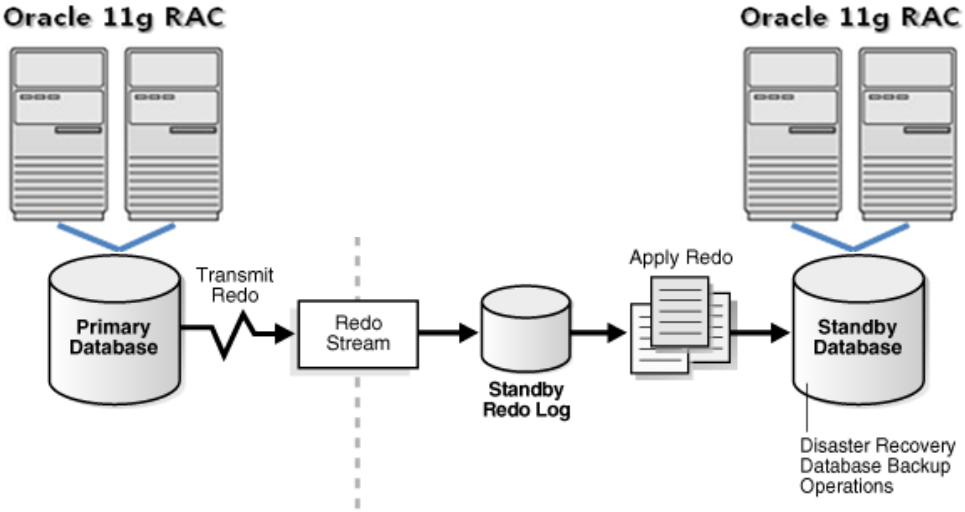
[Active Data Guard with Multiple Standby Databases]



[Active Data Guard with Oracle RAC Standby Databases and a Single Primary]



[Transmitting Redo Data from a Multi-Instance Primary Database]



[Active Data Guard with Oracle RAC Standby Databases and a Multi-Instance Primary]

1.2.1. 특이사항

- Archive destination : MRP(Media Recover Process)가 RAC 시스템의 한쪽 노드에서만 Redo 를 적용할 수 있으므로 양쪽 Node 에서 arhive destination 를 IBM 의 GPFS 와 같은 공유 File system 으로 구성해야 한다. 그렇지 않으면 Data Guard Switch over 나 Failover 시 Archive file 을 옮겨줘야 하는 번거로움이 발생한다.
- Standby Database 에서 MRP Process 의 Node 변경시 수동으로 적용하여야 한다.

2. Oracle Data Guard 11g의 새로운 기능

2.1. Snapshot Standby

Physical Standby Database 부터 생성되는 새로운 유형의 Standby Database 입니다.
일단 생성되면 Snapshot Standby 는 읽기-쓰기가 지원되어 테스트 와 다른용도를 위해 Primary Database 와 독립적인 트랜잭션을 처리할 수 있습니다.
Snapshot Standby 데이터베이스가 Primary Database 로부터 계속해서 업데이트를 받고 아카이빙을 수행하지만 Primary Database 에서 받는 redo 데이터는 Snapshot Standby 가 다시 Physical Standby 데이터베이스로 전환되고 Snapshot Standby Mode 중에 이루어졌던 모든 업데이트가 버려질 때까지 적용되지 않습니다.
snapshot standby database 는 physical standby database 가 snapshot standby database 로 변환된 것에 의해 생성된 모든 업데이트가 가능한 standby database 입니다.
snapshot standby database 는 primary database 에서 redo data 를 받고 archive 하지만 apply 는 하지 않습니다.
snapshot standby database 에서 발생된 모든 local update 를 버린 이후에 일단 snapshot standby database 는 physical standby database 로 전환된 후 primary database 에서 받은 redo data 가 적용됩니다.
snapshot standby database 는 EM 에서 나 Data Guard Broker command line 인 DGMGRL 그리고 SQL*Plus 를 통해서 생성 할 수 있습니다.

2.2. SYNC 와 ASYNC redo 전송의 자동 Failover

Data Guard 10g Release 2 는 새로운 기능인 최대 가용성 보호 모드 (SYNC)의 패스트 스타트 페일오버 (Fast-Start Failover)를 사용하는 자동 페일오버를 도입했습니다. Data Guard 11g 은 패스트 스타트 페일오버를 확장해 자동 페일오버로 인해 원하는 RPO (Recovery Point Objective)를 초과하는 데이터 손실로 이어지지 않도록 하는 사용자 구성 데이터 손실 한계를 추가하는 방식으로 최대 성능 모드 (ASYNC)를 지원합니다.
사용자는 페일오버가 패스트 스타트 페일오버 한계 시간이 지정된 점검 상태나 원하는 ORA 오류에 기초해 소멸될 때까지 기다릴 필요 없이 즉시 발생하도록 자동 페일오버를 구성할 수 있습니다.
새로운 DBMS_DG PL/SQL 패지지를 사용해 애플리케이션이 패스트 스타트 페일오버 옵저버 프로세스를 통보해 자동 페일오버를 시작할 수 있습니다.
성능 강화: 다음과 같은 다양한 성능 강화를 제공합니다.

- 병렬 미디어 복구 (물리적 스탠바이)를 사용해 모든 작업 프로파일에 있어서 물리적 스탠바이 적용 성능을 크게 강화합니다.
- SQL Apply 강화 (논리적 스탠바이)를 사용하면, 구획이 구분이 되지 않고 LOB, LONG, XML 유형의 컬럼을 포함하지 않는 테이블의 삽입 및 업데이트 적용 성능을 증대할 수 있습니다.
- ASYNC redo 전송 강화로 네트워크 지연이 제거되면서 네트워크 작업량이 증대할 것이며, 특히 WAN 구축에서 효과가 뛰어납니다.
- 패스트 스타트 페일오버 사용시 페일오버 시간이 추가로 감소합니다.

2.3. Transient Logical Standby 롤링업그레이드 지원

사용자들은 물리적 스탠바이를 일시 논리적 스탠바이 데이터베이스로 전환해 롤링 데이터베이스 업그레이드에 영향을 미치고, 업그레이드가 완료된 후에 물리적 스탠바이로 다시 전환할 수 있습니다 (KEEP IDENTITY 절사용). 따라서 논리적 스탠바이 데이터베이스를 생성하는데 필요한 중복 스토리지 투자를 하지 않고 롤링데이터베이스 업그레이드를 실행하고자 하는 물리적 스탠바이 사용자들에게 혜택을 제공합니다.

2.4. 데이터보호 강화

물리적 스탠바이는 데이터 손상을 가져올 수 있는 결함이 있는 스토리지 하드웨어와 펌웨어로 인해 발생하는 손실 데이터 파일 쓰기를 감지합니다. Data Guard 는 스탠바이상의 블록 버전을 유입 redo 스트림 버전과 비교합니다. 버전에 차이가 있을 경우, 쓰기 손실을 표시합니다. 사용자는 스탠바이 데이터베이스에 페일오버를 실시하고 데이터 일관성을 복구합니다.

2.5. 보안 강화(SSL 인증 REDO 전송)

SSL 인증을 비밀번호 파일과 함께 사용해 redo 전송을 인증할 수 있습니다. 주: SSL 인증을 위해서는 PKI 인증서, ASO, OID 가 필요합니다.

2.6. 추가 SQL Apply 데이터 유형 지원

SQL Apply 는 다음과 같은 추가 데이터 유형, 기타 오라클 기능, PL/AQL 을 지원합니다.

- XMLType 데이터 유형 (CLOB 로 저장되는 경우)
- 논리적 스탠바이 데이터베이스에서 병렬로 DDL 을 실행하는 기능
- TDE (Transparent Data Encryption)
- DBMS_FGA (Fine Grained Auditing)
- DBMS_RLS (Virtual Private Database: 가상 개인 데이터베이스)

2.7. SQL Apply 관리 강화

DBMS_SCHEDULER 패키지를 사용해 스탠바이 데이터베이스에서 Scheduler Job 을 생성하고 의도된 대로 작동할 수 있도록 적합한 데이터베이스 역할과 연관시킬 수 있습니다 (데이터베이스가 기본인지 스탠바이인지 둘 다인지 등).

Oracle RAC 데이터베이스와 함께 SQL Apply 를 사용하면 각 Oracle RAC 클러스터의 최초 인스턴스를 제외한 모든 인스턴스의 사전 중지 필요성이 더 이상 없습니다. SQL Apply 를 다시 시작할 필요 없이 Data Guard SQL Apply 파라미터를 동적으로 설정할 수도 있습니다. DBMS_LOGSTDBY.APPLY_SET 패키지를 사용해 초기화 파라미터를 동적으로 설정함으로써 논리적 스탠바이 구성의 관리, 업타임, 자동화를 개선할 수 있습니다.

2.8. Data Guard Broker 강화

다음과 같은 기능 강화로 Data Guard Broker 를 사용할 때 관리가 더욱 간편해집니다.

- redo 전송 옵션 지원을 개선해 관리자가 redo 전송 서비스를 위한 연결 설명을 구체화
- 보호 모드와 최대 가용성/최대 성능 사이의 변경 시 데이터베이스 다운타임을 제거
- Oracle Clusterware 를 콜드 페일오버 클러스터로 사용해 고가용성 구성을 위한 단일 인스턴스 데이터베이스 지원

2.9. Enterprise Manager Grid Control 11g 기능 개선

Enterprise Manager 는 다음과 같은 부문의 관리를 한층 간편하게 만들어 줍니다.

- 기존 RMAN 백업에서 스탠바이 데이터베이스 생성
- Oracle RAC Primary Database 에서 Oracle RAC 스탠바이 데이터베이스 생성
- 리포팅, 개발, 테스트를 위한 자동 재기 복사
- 스위치오버나 페일오버시 Enterprise Manager 업무와 성능 한계를 새로운 Primary Database 로 자동 전파
- 패스트 스타트 페일오버를 위한 작동 대체 옵저버 (Fault-tolerant observer)
- Enterprise Manager Data Recovery Advisor 가 IDR (Intelligent Data Repair)를 권장을 할 때 사용 가능한 스탠바이 데이터베이스 활용

2.10. 새로운 Oracle Database 11g 데이터베이스 옵션

2.10.1. Oracle Active Data Guard

프로덕션 데이터베이스에서 한 개 이상의 동기화된 데이터베이스로 자원 집중 활동을 오프 로딩하는 방식으로 QoS 를 강화하는 Oracle Database 11g Enterprise Edition 을 위한 옵션입니다. Active Data Guard 옵션과 함께 제공되는 실시간 쿼리 기능을 사용해 계속해서 프로덕션 데이터베이스에서 받은 변경내용을 Apply 하면서, 쿼리, 소팅, 리포팅, 웹 기반 접속 등을 위한 물리적 스탠바이 데이터베이스에 읽기 전용 접속을 실행할 수 있습니다. 또한 물리적 스탠바이상에서 RMAN 블록 변경 추적을 실행해 물리적 스탠바이 데이터베이스에서 신속한 증가 백업을 실시할 수 있습니다. 테스트에 따르면 RMAN 블록 변경 추적을 사용해 데이터베이스에서 변경 속도를 중간으로 하여 점차로 백업을 증가하면 기존 증가 백업에 비해 최대 20 배 빠르게 완료할 수 있는 것으로 나타났습니다.

2.10.2. Oracle Advanced Compression

(2 년마다 3 배 속도로) 늘어나는 데이터 양을 비용 효율적으로 관리하도록 하는 Oracle Database 11g Enterprise Edition 의 옵션입니다. Advanced Compression 은 네트워크 트래픽과 백업 프로세스 중의 데이터뿐만 아니라 문서, 이미지, 멀티미디어와 같은 다양한 체계/비체계화 된 데이터를 비롯한 모든 유형의 데이터를 압축합니다.

Advanced Compression 옵션은 스탠바이 데이터베이스에서 아카이브 로그 갭의 Data Guard 11g 해결 중에 redo 데이터의 네트워크 압축을 수행합니다. 따라서 네트워크나 스탠바이 데이터베이스 작동 중지 후에 스탠바이 데이터베이스의 재동기화를 가속화하고 네트워크 대역폭을 좀더 효율적으로 활용할 수 있습니다.

3. Oracle 11gR2 RAC SW 설치

GUARD 구성을 위한 특별한 설치 option 은 없으며 굿어스 **기술노트 47 회 11gR2_RAC_Guide 설치과정** 동일하게 진행합니다.

3.1. Primary DB & Standby DB Information

NAME	DBNAME	Instance Name	hostname	Version	Protection mode
Primary DB	GOODUS	GOODUS1	goodus1	11.2.0.1.1	Maximum Performance (LGWR ASYNC)
Primary DB	GOODUS	GOODUS2	goodus2	11.2.0.1.1	Maximum Performance (LGWR ASYNC)
Standby DB	STBY	STBY1	goodus3	11.2.0.1.1	Maximum Performance (LGWR ASYNC)
Standby DB	STBY	STBY2(	goodus4	11.2.0.1.1	Maximum Performance (LGWR ASYNC)

3.2. Oracle 11gR2 parameter 변경 사항

Oracle 11g R2 에서는 Dataguard(이하 DG)관련 파라미터 중 다음 파라미터가 deprecated 됨

standby_archive_dest

4. Oracle 11g RAC Data Guard 구성

4.1. Tnsnames.ora 설정

[PRIMARY TNSNAMES.ORA]

```
STBY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.10.10.111)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.10.10.112)(PORT = 1521))
    (FAIL_OVER=ON)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = STBY)
    )
  )
```

[STANDBY TNSNAMES.ORA]

```
GOODUS =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.10.10.101)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.10.10.102)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = GOODUS)
    )
  )
```

4.2. Database Force logging setup

(oracle 유저로 primary 1 번 노드에서 수행)

```
[oracle@goodus1:/oracle/db]$ srvctl stop database -d GOODUS
```

```
SQL> Startup Mount
SQL> ALTER DATABASE FORCE LOGGING;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
SQL> SELECT FORCE_LOGGING,SUPPLEMENTAL_LOG_DATA_MIN FROM V$DATABASE;
FOR   SUPPLEME
------   --------
YES   YES
```

4.3. Standby Redo log 생성

Oracle 사용자로 Primary 에서 수행합니다.

Standby Redo Logfile 은 Primary db log group 보다 1 개 더 많게 구성 할 것을 권고 하며 반드시 Primary redo log 와 Standby redo log 가 같은 크기를 가지고 있어야 합니다. 약 다를 경우에는 차후에 ORA-16139 media recovery required 에러가 발생하면서 switchover 나 failover 가 정상적으로 수행되지 않을 수 있습니다.

[PRIMAY]

```
- ================================================
- Standby redolog 구성 (primary redo log 보다 1 개 많은 6 개 group 구성)
- ================================================

        ALTER DATABASE ADD STANDBY LOGFILE THREAD 1
        '/dev/rcoredo15_lv'   SIZE 700M,
        '/dev/rcoredo16_lv'   SIZE 700M,
        '/dev/rcoredo17_lv'   SIZE 700M,
        '/dev/rcoredo18_lv'   SIZE 700M,
        '/dev/rcoredo19_lv'   SIZE 700M,
        '/dev/rcoredo20_lv'   SIZE 700M,
        '/dev/rcoredo21_lv'   SIZE 700M,
        '/dev/rcoredo22_lv'   SIZE 700M;

        ALTER DATABASE ADD STANDBY LOGFILE THREAD 2
        '/dev/rcoredo23_lv' SIZE 700M,
        '/dev/rcoredo24_lv' SIZE 700M,
        '/dev/rcoredo25_lv' SIZE 700M,
        '/dev/rcoredo26_lv' SIZE 700M,
        '/dev/rcoredo27_lv' SIZE 700M,
        '/dev/rcoredo28_lv' SIZE 700M,
        '/dev/rcoredo29_lv' SIZE 700M,
         '/dev/rcoredo30_lv' SIZE 700M;
```

4.4. PRIMARY/STANDBY Init parameter 설정

[PRIMARY]

```
- ====================================
- SPFILE 을 PFILE 로 변환
- ====================================

SQL> create pfile='/oracle/db/initGOODUS.ora' from spfile;

SQL> !vi /oracle/db/initGOODUS.ora

##############################
##   primary role parameters      ##
##############################

*.db_unique_name='GOODUS'

GOODUS1.log_archive_config='dg_config=(GOODUS,STBY)'
GOODUS2.log_archive_config='dg_config=(GOODUS,STBY)
'
*.log_archive_dest_1='location=/archive mandatory'

GOODUS1.log_archive_dest_2='SERVICE=STBY LGWR ASYNC VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)
DB_UNIQUE_NAME=STBY'
GOODUS2.log_archive_dest_2='SERVICE=STBY LGWR ASYNC VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)
DB_UNIQUE_NAME=STBY'
*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'

*.standby_file_management='auto'
➔ MRP 프로세스가 한쪽에서 redo 를 apply 하므로 log_archive_dest_2 의 서비스를 goodus3 의 /arch1 으로
  떨어지도록 해야함. GOODUS2. log_archive_dest_2 를 goodus4 로 서비스하면 goodus2 에서 v$archive_dest 의
  Status 조회시 dest2 에서 ORA-16191 가 발생한다.

##############################
##   standby role parameters      ##
##############################
GOODUS1.fal_client='GOODUS1'
GOODUS2.fal_client='GOODUS2'
GOODUS1.fal_server='STBY1'
GOODUS2.fal_server='STBY2'

- ====================================
- PFILE 을 SPFILE 로 변환
- ====================================

SQL> !cat $ORACLE_HOME/dbs/initGOODUS1.ora
SPFILE='/dev/rspfile_lv'                    # line added by Agent

SQL> create SPFILE='/dev/rspfile_lv' from pfile='/oracle/db/initGOODUS.ora';
```

[STANDBY]

```
- ====================================
- SPFILE 을 PFILE 로 변환
- ====================================
```

```
SQL> create pfile='/oracle/db/initSTBY.ora' from spfile;

SQL> !vi /oracle/db/initSTBY.ora

################################
##   primary role parameters      ##
################################

*.db_name='GOODUS'
*.db_unique_name='STBY'

*.log_archive_config='dg_config=(GOODUS,STBY)'

STBY1.log_archive_dest_1='LOCATION=/arch mandatory'
STBY2.log_archive_dest_1='LOCATION=/arch mandatory'

STBY1.log_archive_dest_2='SERVICE=GOODUS LGWR ASYNC VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=GOODUS'
STBY2.log_archive_dest_2='SERVICE=GOODUS LGWR ASYNC VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=GOODUS'

*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'


################################
##   standby role parameters      ##
################################
*.fal_client='STBY'
*.fal_server='GOODUS'
*.db_file_name_convert='/dev/r','/dev/rk'
*.log_file_name_convert='/dev/r','/dev/rk'
*.standby_file_management=auto
➔ Primary 와 Standby RAC 시스템의 각각의 Node 가 동일한 Archive dest 를 사용할수 있도록 IBM 의 GPFS 와 같은 공유 File system 으로 구성해야
```

4.5. Standby 용 controlfile 생성

[PRIMARY –GOODUS1]

```
Sql> alter database create standby controlfile as '/oracle/db/stby_control.ctl';
Database altered.
[oracle@goodus1:/oracle/db]$   dd if=/oracle/db/stby_control.ctl bs=512 | ssh goodus3 "dd of=/dev/rkcoredo39_lv
bs=512"
[oracle@goodus1:/oracle/db]$   dd if=/oracle/db/stby_control.ctl bs=512 | ssh goodus3 "dd of=/dev/rkcoredo40_lv
bs=512"
```

4.6. Standby 파일 위치 directory 생성

[STANDBY –STBY1]

Log , archive, datafile 위치 directory 생성

4.7. Stand by 서버로 파일 이전

[PRIMARY –GOODUS1]

```
- ====================================
- primary 의 패스워드 파일을 standby 서버로 이전
- ====================================

1 번 노드에서 수행
Scp $ORACLE_HOME/dbs/orapwGOODUS1 goodus3:/oracle/db/11g/dbs/

1 번 노드에서 수행
Scp $ORACLE_HOME/dbs/orapwGOODUS1 goodus4:/oracle/db/11g/dbs/

**패스워드 파일이 동일 하지 않을 경우 ORA-01031: insufficient privileges 에러가 발생하며 redo 전송이 되지 않음

SQL>select dest_id,status,error from v$archive_dest;
DEST_ID STATUS                ERROR
---------- ----------------- ----------------------------------------------------------------
         1 VALID
         2 INVALID     ORA-01031: insufficient privileges

- ====================================
- datafile ,redo log,standby 용 controlfile standby 서버로 이전
- ====================================
→primary db shutdown 후 모든 파일 이전
```

4.8. PRIMARY DB OPEN

[PRIMARY –GOODUS1]

```
[oracle@goodus1:/oracle/db]$ srvctl start database –d GOODUS


SQL> select DATABASE_ROLE,PROTECTION_MODE from v$database;

DATABASE_ROLE      PROTECTION_MODE

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

PRIMARY            MAXIMUM PERFORMANCE
```

4.9. STANDBY DB OPEN

[STANDBY –STBY1]

```
SQL> startup mount

SQL>alter database recover managed standby database disconnect from session;   →start redo apply

SQL>alter database recover managed standby database cancel;

SQL>alter database open;

SQL>alter database recover managed standby database using current logfile disconnect;

→real-time redo apply


SQL> select open_mode from v$database;
```

```
OPEN_MODE
--------------------
READ ONLY WITH APPLY
```

[STANDBY –STBY2]

```
SQL> startup

SQL> select open_mode from v$database;

OPEN_MODE
--------------------
READ ONLY WITH APPLY
```

4.10. STANDBY LOGFILE ADD

[STANDBY –STBY1]

```
SQL> STARTUP MOUNT;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1
            '/dev/rk2coredo15_lv'   SIZE 700M,
            '/dev/rk2coredo16_lv'   SIZE 700M,
            '/dev/rk2coredo17_lv'   SIZE 700M,
            '/dev/rk2coredo18_lv'   SIZE 700M,
            '/dev/rk2coredo19_lv'   SIZE 700M,
            '/dev/rk2coredo20_lv'   SIZE 700M,
            '/dev/rk2coredo21_lv'   SIZE 700M,
            '/dev/rk2coredo22_lv'   SIZE 700M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2
            '/dev/rk2coredo23_lv' SIZE 700M,
            '/dev/rk2coredo24_lv' SIZE 700M,
            '/dev/rk2coredo25_lv' SIZE 700M,
            '/dev/rk2coredo26_lv' SIZE 700M,
            '/dev/rk2coredo27_lv' SIZE 700M,
            '/dev/rk2coredo28_lv' SIZE 700M,
            '/dev/rk2coredo29_lv' SIZE 700M,
           '/dev/rk2coredo30_lv' SIZE 700M;
```

4.11. CRS에 Service Register

CRS 에 Database 와 service 등을 register 할때는 PHYSICAL_STANDBY 옵션을 추가해줘야 장애시 Failover 에 이상이 발생하지 않는다.

 # Add Database Syntax

Usage: srvctl add database -d <db_unique_name> -o <oracle_home> [-m <domain_name>] [-p <spfile>] [-r {PRIMARY | PHYSICAL_STANDBY | LOGICAL_STANDBY | SNAPSHOT_STANDBY}] [-s <start_options>] [-t <stop_options>] [-n <db_name>] [-y {AUTOMATIC | MANUAL}] [-g "<serverpool_list>"] [-x <node_name>] [-a "<diskgroup_list>"]

[oracle@goodus3:/oracle/db]$ **srvctl add database -d STBY -o /oracle/db/11g -p /dev/rspfile_lv -r PHYSICAL_STANDBY -n GOODUS**

[oracle@goodus3:/oracle/db]$ **srvctl config database -d STBY**

Database unique name: STBY

Database name: GOODUS

Oracle home: /oracle/db/11g

Oracle user: oracle

Spfile: /dev/rspfile_lv

Domain:

Start options: open

Stop options: immediate

Database role: **PHYSICAL_STANDBY**

Management policy: AUTOMATIC

Server pools: STBY

Database instances: STBY1,STBY2

Disk Groups:

Services:

Database is administrator managed

[oracle@goodus3:/oracle/db]$

 # Add Service Syntax

Usage: srvctl add service -d <db_unique_name> -s <service_name> {-r "<preferred_list>" [-a "<available_list>"] [-P {BASIC | NONE | PRECONNECT}] | -g <server_pool> [-c {UNIFORM | SINGLETON}] } [-k <net_num>] [-l [PRIMARY][,PHYSICAL_STANDBY][,LOGICAL_STANDBY][,SNAPSHOT_STANDBY]] [-y {AUTOMATIC | MANUAL}] [-q {TRUE|FALSE}] [-x {TRUE|FALSE}] [-j {SHORT|LONG}] [-B {NONE|SERVICE_TIME|THROUGHPUT}] [-e {NONE|SESSION|SELECT}] [-m {NONE|BASIC}] [-z <failover_retries>] [-w <failover_delay>]

[oracle@goodus3:/oracle/db]$ **srvctl add service -d STBY –s TESTSVC –r STBY1 –a STBY2 –l PHYSICAL_STANDBY**

[oracle@kdb41:/oracle/db]$ srvctl config service -d STBY -s testsvc

Service name: TESTSVC

```
Service is enabled
Server pool: STBY_TESTSVC
Cardinality: 1
Disconnect: false
Service role: PHYSICAL_STANDBY
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Failover type: NONE
Failover method: NONE
TAF failover retries: 0
TAF failover delay: 0
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
Preferred instances: STBY1
Available instances: STBY2
[oracle@goodus3:/oracle/db]$
```

4.12. Oracle 11g Data Guard 구성 점검

4.12.1. Redo 전송 확인

[PRIMARY-GOODUS1]

```
Col error for a30
select dest_id,status,error from v$archive_dest where rownum   < 3;
DEST_ID STATUS      ERROR
---------- --------- ------------------------------
         1 VALID
         2 VALID
```

→ redo 전송이 안될 경우 위의 쿼리로 error 원인을 파악 합니다.
TNS 오류 발생시 리스너 및 tnsping 을 확인합니다.

4.12.2. Standby recovery 확인

[STANDBY-STBY1]

- 오라클 백그라운드 프로세스 중 recovery 를 담당하는 mrp 프로세스가 떠 있는지 확인합니다.
- 현재 구성은 Standby 쪽도 RAC 이기 때문에 mrp 프로세스는 recover managed 명령을 수행한 노드에서만 구동됩니다. 아래의 결과를 보면, 1 번 노드에서만 mpr 프로세스가 구동된 것을 확인할 수 있습니다.

```
[oracle@goodus3:/oracle/db]$ ps -ef | grep mrp
 oracle 188620      1    0 10:34:04      -   0:00 ora_mrp0_STBY1
 oracle 459222  73762    0 11:26:46  pts/0   0:00 grep mrp

[oracle@goodus4:/oracle/db]$ ps -ef | grep mrp
oracle 213438 569636    0 11:27:07  pts/0   0:00 grep mrp
```

- alert log 확인

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT
Attempt to start background Managed Standby Recovery process (STBY1)
Fri Apr 16 10:34:04 2010
MRP0 started with pid=40, OS id=188620
MRP0: Background Managed Standby Recovery process started (STBY1)
```

-Mrp 프로세스가 start 되지 않았다면 alert log 를 통해 원인을 확인 하고 조취를 취한 후 redo apply 를 재시작 합니다.

--redo apply start

```
SQL>alter database recover managed standby database disconnect from session;
```

--redo apply stop

```
SQL> alter database recover managed standby database cancel;
```

4.12.3. Redo data archiving 확인

[ACTIVE-GOODUS1]

```
--현재 archived file 확인
SQL> SELECT * FROM (SELECT NAME, THREAD#, SEQUENCE#, FIRST_TIME, APPLIED
       FROM V$ARCHIVED_LOG ORDER BY SEQUENCE# DESC)
       where rownum < 10;

NAME                                               THREAD#  SEQUENCE# FIRST_TIME       APPLIED
-------------------------------------------------- ---------- ---------- --------------- ---------
STBY1                                                       1       8191 16-APR-10       YES
/archive/arch_1_713988964_8191.arc                          1       8191 16-APR-10       NO
STBY1                                                       1       8190 16-APR-10       YES
/archive/arch_1_713988964_8190.arc                          1       8190 16-APR-10       NO
STBY1                                                       1       8189 16-APR-10       YES
/archive/arch_1_713988964_8189.arc                          1       8189 16-APR-10       NO
STBY1                                                       1       8188 16-APR-10       YES
/archive/arch_1_713988964_8188.arc                          1       8188 16-APR-10       NO
STBY1                                                       1       8187 16-APR-10       YES--force logging
```

[STANDBY-STBY1]

```
--새로운 redo data 가 archiveing 되었는지 확인
SQL> SELECT * FROM (SELECT NAME, THREAD#, SEQUENCE#, FIRST_TIME, APPLIED
       FROM V$ARCHIVED_LOG ORDER BY SEQUENCE# DESC)
```

```
      where rownum < 10;

NAME                                     THREAD#  SEQUENCE# FIRST_TIME   APPLIED
---------------------------------------- ---------- ---------- --------------- ---------
/arch1/arch_1_713988964_8191.arc              1       8191 16-APR-10       YES
/arch1/arch_1_713988964_8190.arc              1       8190 16-APR-10       YES
/arch1/arch_1_713988964_8189.arc              1       8189 16-APR-10       YES
/arch1/arch_1_713988964_8188.arc              1       8188 16-APR-10       YES
/arch1/arch_1_713988964_8187.arc              1       8187 16-APR-10       YES
/arch1/arch_1_713988964_8186.arc              1       8186 16-APR-10       YES
/arch1/arch_1_713988964_8185.arc              1       8185 16-APR-10       YES
/arch1/arch_1_713988964_8184.arc              1       8184 16-APR-10       YES
/arch1/arch_1_713988964_8183.arc              1       8183 16-APR-10       YES
```

STANDBY DATABASE Restart

Standby Database 재 시작시 에는 아래와 같이 Real time apply 를 시작해 줘야 함.

```
[oracle@goodus3:/oracle/db]$ srvctl start database -d STBY
SQL> alter database recover managed standby database using current logfile disconnect;
```

구성시 Error 조치 방법

[Standby Alert.log]

ARC5: Archiving not possible: error count exceeded

ARCH: Archival stopped, error occurred. Will continue retrying

ORACLE Instance STBY1 - Archival Error

ORA-16014: log 11 sequence# 8179 not archived, no available destinations

ORA-00312: online log 11 thread 1: '/dev/rk2coredo11_lv'

Errors in file /oracle/db/diag/rdbms/STBY/STBY1/trace/STBY1_arc5_245872.trc:

ORA-16014: log 11 sequence# 8179 not archived, no available destinations

ORA-00312: online log 11 thread 1: '/dev/rk2coredo11_lv'

조치:

standby_archive_dest 파라미터 설정시 제거 (deprecated 됨)

log_archive_dest_1 파라미터에 valid for 옵션 설정시 제거

4.13. Trouble Shooting

4.13.1. Standby redo log file 의 삭제와 재생성

Standby Database 의 Datafile 과 Redo log file 은 Primary 와의 Switch over 를 위해서 File name 을 Primary DB 와 다르게 구성하여야 한다. 특히 log_file_name_convert 의 경 DB Startup 시 **ORA-19527: physical standby redo log must be renamed** 에러가 발생한다. db_file_name_convert parameter 의 경우는 "#"로 주석처리하거나 삭제하면 되지만, log_file_name_convert 는 필수 parameter 이므로 아래와 같이 조치하면 된다.

```
#*.db_file_name_convert='/dev/r','/dev/rk'
*.log_file_name_convert='/dev/r','/dev/r'
```

그러나 Data Guard 의 Switch over 를 위해서라면 file name 을 rename 해주는것이 좋다. Rename 전에 이미 Data Guard DB 를 Open 하여 운영하였다면, 다음과 같이 Standby Redo Log 의 file name 을 변경하여야 한다.

** OS 에서 Volume rename 후 작업

```
alter database rename file   '/dev/rcoredo15_lv' to   '/dev/rkcoredo15_lv' ;
alter database rename file   '/dev/rcoredo16_lv' to   '/dev/rkcoredo16_lv' ;
alter database rename file   '/dev/rcoredo17_lv' to   '/dev/rkcoredo17_lv' ;
alter database rename file   '/dev/rcoredo18_lv' to   '/dev/rkcoredo18_lv' ;
alter database rename file   '/dev/rcoredo19_lv' to   '/dev/rkcoredo19_lv' ;
alter database rename file   '/dev/rcoredo20_lv' to   '/dev/rkcoredo20_lv' ;
alter database rename file   '/dev/rcoredo21_lv' to   '/dev/rkcoredo21_lv' ;
alter database rename file   '/dev/rcoredo22_lv' to   '/dev/rkcoredo22_lv' ;
alter database rename file   '/dev/rcoredo23_lv' to   '/dev/rkcoredo23_lv' ;
alter database rename file   '/dev/rcoredo24_lv' to   '/dev/rkcoredo24_lv' ;
alter database rename file   '/dev/rcoredo25_lv' to   '/dev/rkcoredo25_lv' ;
alter database rename file   '/dev/rcoredo26_lv' to   '/dev/rkcoredo26_lv' ;
alter database rename file   '/dev/rcoredo27_lv' to   '/dev/rkcoredo27_lv' ;
alter database rename file   '/dev/rcoredo28_lv' to   '/dev/rkcoredo28_lv' ;
alter database rename file   '/dev/rcoredo29_lv' to   '/dev/rkcoredo29_lv' ;
alter database rename file   '/dev/rcoredo30_lv' to   '/dev/rkcoredo30_lv' ;
```

특별히 각 Thread 별로 사용중인 stby redo log file 이라면 Rename 되지 않고 ORA-01621 cannot rename member of current log if database is open 에러가 발생한다.

```
SQL> SELECT GROUP#, STATUS FROM V$LOG;
GROUP# STATUS
---------- ----------------
1 CLLEARING
3 CLEARING
2 CLEARING
..
16 CLEARING_CURRENT
..
22 CLEARING
```

23 CLEARING_CURRENT

이 것은 Online 중이여서 Drop 을 시도하게 되면 ORA-01623 log string is current log for instance string (thread string) - cannot drop 에러가 발생하게 된다.

이런 상황이라면 다음과 같이 조치한다.

1. Stop Redo Apply
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

2. Set STANDBY_FILE_MANAGEMENT to MANUAL.
SQL> ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT='MANUAL';

3. Clear the Online Redo Logfile Group
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 16;
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 23;

4. Drop the Online Redo Logfile Group
SQL>ALTER DATABASE DROP LOGFILE GROUP 16;
SQL>ALTER DATABASE DROP LOGFILE GROUP 23;

만약에 step 3 을 생략하였을 경우 ORA-01624 while droping the Online Redolog Group with Status CLEARING 가 발생할 수 있다.

5. Standby redo log 재생성
SQL> ALTER DATABASE ADD STANDBY LOGFILE '/dev/rkcoredo16_lv' SIZE 700M
BLOCKSIZE 512 REUSE;
SQL> ALTER DATABASE ADD STANDBY LOGFILE '/dev/rkcoredo23_lv' SIZE 700M
BLOCKSIZE 512 REUSE;

6. Set STANDBY_FILE_MANAGEMENT to AUTO
SQL> ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT='AUTO';

7. Start Redo Apply:
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;

6. init parameter 변경
*.log_file_name_convert='/dev/r','/dev/rk'

4.13.2. Standby db start시 Temporary tablespace 재생성 안되는 문제

기본적으로 Guard 에서 temporary tablespace 는 primary 와 stby 가 동기화 되지 않는다. 아래 메시지는 이미 dictionary 에는 temporary tablespace 가 등록되어있는데 실제 raw device 손실로 db open 시 temp 메카니즘인 재생성이 안되는 문제이다. 이것은 dd if=/dev/zero 로 빈 raw device 를 만들어도 마찬가지 경우이므로 database open(real time apply 중)후 해당 tempfile 메뉴얼하게 삭제후 add temp 하면된다.

**** 에러 메시지**

File 1004 not verified due to error ORA-01122
Read of datafile '/dev/rkdata014_lv' (fno 1005) header failed with ORA-01210
Hex dump of (file 1005, block 1) in trace file
/oracle/db/diag/rdbms/stby/STBY1/trace/STBY1_dbw0_336322.trc
Corrupt block relative dba: 0x00c00001 (file 1005, block 1)
Completely zero block found during datafile header read
Rereading datafile 1005 header failed with ORA-01210
Hex dump of (file 1005, block 1) in trace file
/oracle/db/diag/rdbms/stby/STBY1/trace/STBY1_dbw0_336322.trc
Corrupt block relative dba: 0x00c00001 (file 1005, block 1)
Completely zero block found during datafile header read
Errors in file /oracle/db/diag/rdbms/stby/STBY1/trace/STBY1_dbw0_336322.trc:
ORA-01186: file 1005 failed verification tests
ORA-01122: database file 1005 failed verification check
ORA-01110: data file 1005: '/dev/rkdata014_lv'
ORA-01210: data file header is media corrupt
File 1005 not verified due to error ORA-01122
Read of datafile '/dev/rkdata017_lv' (fno 1006) header failed with ORA-01210
Hex dump of (file 1006, block 1) in trace file
/oracle/db/diag/rdbms/stby/STBY1/trace/STBY1_dbw0_336322.trc
Corrupt block relative dba: 0x00400001 (file 1006, block 1)
Completely zero block found during datafile header read
Rereading datafile 1006 header failed with ORA-01210
Hex dump of (file 1006, block 1) in trace file
/oracle/db/diag/rdbms/stby/STBY1/trace/STBY1_dbw0_336322.trc
Corrupt block relative dba: 0x00400001 (file 1006, block 1)
Completely zero block found during datafile header read
Errors in file /oracle/db/diag/rdbms/stby/STBY1/trace/STBY1_dbw0_336322.trc:
ORA-01186: file 1006 failed verification tests
ORA-01122: database file 1006 failed verification check
ORA-01110: data file 1006: '/dev/rkdata017_lv'

ORA-01210: data file header is media corrupt
File 1006 not verified due to error ORA-01122
Dictionary check complete
Cannot re-create tempfile /dev/rkdata012_lv, the same name file exists
Cannot re-create tempfile /dev/rkdata013_lv, the same name file exists
Cannot re-create tempfile /dev/rkdata014_lv, the same name file exists
Cannot re-create tempfile /dev/rkdata015_lv, the same name file exists
Cannot re-create tempfile /dev/rkdata016_lv, the same name file exists
Cannot re-create tempfile /dev/rkdata017_lv, the same name file exists
Database Characterset is UTF8
No Resource Manager plan active
Starting background process GTX0

** 조치방법

- Temp file drop

```
alter database tempfile '/dev/rkdata012_lv' drop;
alter database tempfile '/dev/rkdata013_lv' drop;
alter database tempfile '/dev/rkdata014_lv' drop;
alter database tempfile '/dev/rkdata015_lv' drop;
alter database tempfile '/dev/rkdata016_lv' drop;
alter database tempfile '/dev/rkdata017_lv' drop;
```

- Temp file add

```
ALTER TABLESPACE TEMP_KAD ADD TEMPFILE '/dev/rdata016_lv'
      SIZE 20735M REUSE AUTOEXTEND OFF;
ALTER TABLESPACE TEMP_KAD ADD TEMPFILE '/dev/rdata015_lv'
      SIZE 20735M REUSE AUTOEXTEND OFF;
ALTER TABLESPACE TEMP_KAD ADD TEMPFILE '/dev/rdata014_lv'
      SIZE 20735M REUSE AUTOEXTEND OFF;
ALTER TABLESPACE TEMP_KIN ADD TEMPFILE '/dev/rdata017_lv'
      SIZE 20735M REUSE AUTOEXTEND OFF;
ALTER TABLESPACE TEMP ADD TEMPFILE '/dev/rdata013_lv'
      SIZE 20735M REUSE AUTOEXTEND OFF;
ALTER TABLESPACE TEMP ADD TEMPFILE '/dev/rdata012_lv'
      SIZE 20735M REUSE AUTOEXTEND OFF;
```

5. 참고문서

- Maximum Availability Architecture - Oracle Active Data Guard Oracle Data Guard 11g Release 1
- Oracle® Data Guard Concepts and Administration 11g Release 2 (11.2)
- Oracle® Grid Infrastructure Installation Guide 11g Release 2 (11.2) for AIX Operating System
- Oracle Database 11g New Features – Availability
- RAC Assurance Support Team: RAC Starter Kit and Best Practices (AIX)