

---

# 11g ASM Improvement

Author	조 훈
Creation Date	2010-08-17
Last Updated	2010-08-17
Version	1.0
Copyright(C) 2004 Goodus Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2010-08-17	조 훈	문서 최초 작성
2			
3			

---

## 목차

1. OVERVIEW .....	3
2. Fast Mirror Resync.....	3
3. ASM Preferred Mirror Read.....	4
4. ASM 확장성과 성능향상 .....	6
5. SYSASM 권한 .....	7
6. ASM 디스크 그룹 호환성 .....	8
7. ASM Disk Group Attributes .....	9
8. Enhanced Disk Group Checks.....	10
9. Restricted Mount Disk Group for Fast Rebalance .....	11
10. Mount Force Disk Group.....	12
11. Forcing Disk Group Drop .....	12
12. ASMCMD Extension in Oracle Database 11g Release 1.....	13
13. ASMCMD Command Extensions.....	14
14. ASM ACLs .....	16
15. SQL 명령어 사용으로 ASM ACLs 관리 .....	18
16. ASMCMD 명령어 사용으로 ASM ACLs 관리 .....	19
17. ASM Intelligent Data Placement .....	20
18. ASM Intelligent Data Placement Best Practices.....	22

---

## 1. OVERVIEW

정보의 중요성이 점점 부각되면서 이 정보를 저장하는 스토리지의 중요성은 두 말할 나위가 없습니다. 이러한 정보들이 시간이 지나면서 기하 급수적으로 늘어나게 되는데 이렇게 증가된 스토리지를 성능, 안정성 및 보안등의 포인트를 최적의 상태로 유지하는 일 또한 매우 중요한 일이 되었습니다.

현재와 같은 대규모 데이터를 다루기 위해서는 DBMS의 특성을 고려하지 않은 스토리지 관점의 접근 방식 보다는 DBMS에 알맞은 공간 할당으로 안정성과 성능을 고려한 구성이 대두되었습니다.

Oracle 11g의 ASM은 추가 비용 없이 데이터베이스 파일에 대한 클러스터 파일 시스템 및 볼륨메니저 기능을 제공하는 Oracle Product입니다. ASM을 통해 스토리지 비용 절감, 스토리지 관리 포인트를 최소화 할 수 있으며 자동 저장공간 관리를 이용하여 최적의 성능을 유지할 수 있습니다.

ASM은 자동 Mirror 디스크에 대한 복구, resynchronization 등을 통하여 지속적인 서비스가 가능하도록 하였습니다.

## 2. Fast Mirror Resync

Fast Mirror Resync가 지원되지 않던 이전 버전에서는 ASM redundancy를 사용시 Disk가 순간 failure(케이블, 전원, host bus adaptor, disk controller의 failure)로 Offline이 되면 ASM은 disk group으로부터 해당 디스크를 drop시키고 할당된 extents를 남아있는 disk에 re-creating합니다.

하지만 ASM Fast Mirror Resync로 일시적인 disk의 failure시 resynchronize에 필요한 시간을 획기적으로 줄였습니다. Disk가 순간 failure로 인하여 Offline되면 ASM은 변경된 extent를 추적하여 disk가 복구된 후 변경된 extent만 resync합니다.

ASM disk path fail시, ASM disk는 offline되지만 DISK\_REPAIR\_TIME에 설정한 값에 해당하는 시간이 지나기 전에는 dropped되지는 않습니다. 이 값이 설정되면 수리를 완료한 후 resynchronize할 수 있을 때까지 ASM은 기다립니다.

- enable fast mirror resync:

```
ALTER DISKGROUP dgroupA SET ATTRIBUTE 'DISK_REPAIR_TIME' = '5H';
```

- manually take the ASM disk offline:

```
ALTER DISKGROUP dgroupA  
OFFLINE DISKS IN FAILGROUP contrl2 DROP AFTER 7H;
```

- bring the disk group back online:

```
ALTER DISKGROUP dgroupA  
ONLINE DISKS IN FAILGROUP contrler2 POWER 2 WAIT;
```

- drop the failure group if it cannot be repaired:

```
ALTER DISKGROUP dgroupA DROP DISKS IN FAILGROUP contrl2 FORCE;
```

현재 속성값은 V\$ASM\_ATTRIBUTE 를 통해 확인 할 수 있습니다.

V\$ASM\_DISK 혹은 V\$ASM\_DISK\_IOSTAT 의 REPAIR\_TIMER 컬럼을 통해 offlined disk 가 drop 되기 전에 남은 시간을 확인 할 수 있습니다.

V\$ASM\_OPERATION 의 OPERATION 컬럼에 disk resync operation 이 나타납니다.

DISK\_REPAIR\_TIME 의 기본 값은 3.6 시간 입니다.

### 3. ASM Preferred Mirror Read

Oracle 10g 에서 ASM failure groups 의 구성은 ASM 은 항상 mirrored extent 의 primary copy 를 읽는다. 특히 확장된 클러스터 구성에서 local copy 로부터 extent 의 읽기가 향상된 성능을 제공한다.

Oracle 11g 에서 ASM\_PREFERRED\_READ\_FAILURE\_GROUPS 의 새 초기화 파라미터를 사용하여 preferred mirror read 를 구성할 수 있다. failure groups 의 disk 는 preferred read disks 가 된다. 그렇게 하여 모든 node 는 로컬 디스크로부터 읽을 수 있다. 이 결과 높은 효율과 성능 그리고 네트워크 부하를 줄일 수 있게 되었다. 이 파라미터는 인스턴스 별로 설정한다.

#### setup

- On first instance

```
ASM_PREFERRED_READ_FAILURE_GROUPS=DATA.SITEA
```

```
ALTER SYSTEM SET ASM_PREFERRED_READ_FAILURE_GROUPS =  
'DATA.DATA_0000' , 'TEST.TEST_0000';
```

- On second instance

```
ASM_PREFERRED_READ_FAILURE_GROUPS=DATA.SITEB
```

```
ALTER SYSTEM SET ASM_PREFERRED_READ_FAILURE_GROUPS =  
'DATA.DATA_0001' , 'TEST.TEST_0001';
```

이 파라미터는 콤마로 구분된 failure group 리스트를 포함한다. 각 failure group 이름은 disk group 이름과 “.” 문자의 접두어로 지정된다. 이 파라미터는 alter system 명령으로 언제든지 수정할 수 있다. 이 파라미터는 ASM 인스턴스에만 유효합니다. 확장된 클러스터에서 failure groups 은 인스턴스에 상응하는 로컬 디스크에 포함된 파라미터로 지정됩니다.

#### Monitor

```

SQL> select instname,
2  dbname,
3  group_number,
4  failgroup,
5  disk_number,
6  reads,
7  writes
8  from v$asm_disk_iostat
9  where group_number in (1,4)
10 order by 1,2,3,4,5,6;

```

INSTNAME	DBNAME	GROUP_NUMBER	FAILGROUP	DISK_NUMBER	READS	WRITES
HAPPY1	HAPPY	1	DATA_0000	0	6191	7703
HAPPY2	HAPPY	1	DATA_0001	1	18224	7703
HAPPY1	HAPPY	4	TEST_0000	0	729	312
HAPPY2	HAPPY	4	TEST_0001	1	446	312

PREFERRED\_READ 컬럼은 v\$ASM\_DISK 에 추가되었습니다. 만약 컬럼의 값이 Y 이면 디스크 그룹이 preferred read failure group 에 관련된 디스크 소유입니다.

ASM preferred read failure groups 의 성능이슈는 v\$ASM\_DISK\_IOSTAT 로 확인 할 수 있습니다. 이 뷰는 각 ASM 클라이언트의 disk input/output 통계를 보여줍니다. 이 뷰는 데이터베이스 인스턴스로 부터 쿼리 됩니다.

위 결과에서 인스턴스 HAPPY1, HAPPY2 의 preferred failgroup 이 각각 DATA\_0000, DATA\_0001 로 설정되어 있는 것을 확인 할 수 있습니다. WRITE 컬럼의 값은 7703 으로 동일하고 이는 쓰기작업이 양쪽 디스크에서 동일하게 수행됨을 의미합니다. READ 컬럼을 보면 인스턴스에 대해 각각 6191, 18224 임을 확인 할 수 있습니다. HAPPY2 인스턴스가 preferred failgroup 인 DATA\_0001 로부터 더 많은 읽기 작업을 수행하였습니다. TEST 그룹의 경우 HAPPY1 인스턴스는 preferred failgroup 인 DATA\_0000 으로 부터 더 많은 읽기 작업을 수행하였습니다.

Preferred Read 는 “스트레치(stretch)”클러스터, 다시 말해 노드 간의 거리가 매우 먼 클러스터 환경에서 특히 유용합니다. 이 설정을 이용하여 읽기 작업을 특정 디스크로 한정함으로써 성능을 개선 할 수 있습니다.

확장 클러스터에서 좋은 disk group 구성의 몇 가지 입니다. 좋은 구성은 확장 클러스터 disk group 의 성능과 가용성을 필요로 합니다.

two-site extended cluster 예를 들면, normal redundancy disk group 는 two failure groups 을 가져야 합니다. 한 개의 사이트에 모든 디스크는 같은 failure group 에 속해집니다. 각 인스턴스에 의해 failure group 우선 읽기가 명시 되어야 하는 것은 한 개의 failure group 입니다. 만약 2 개 이상의 failure groups 이라면 ASM 은 virtual extent 를 상대의 사이트에 mirror 할 지도 모릅니다.

더욱이 만약 2 개 이상의 failure groups 다운된다면 disk group 도 같이 다운될 것이다. 만약 disk group 이 high-redundancy disk group 로 생성 되었다면, 두 개의 failure groups 은 로컬 인스턴스를 위한 우선 읽기 failure groups 가 명시된 로컬 디스크가 각 사이트에 생성되어야 합니다.

three-site extended cluster 예를 들면, high redundancy disk group 는 three failure groups 이 사용 됩니다. 이렇게 하여 ASM 은 각 사이트 로컬에 mirror copy 를 통해 virtual extent 보장하고, 3 개의 사이트 중 아무것이나 disk group 이 비극적인 재해로부터 disk group 은 보호 됩니다

## 4. ASM 확장성과 성능향상

ASM 가변 크기 extents 는 ASM 메모리 사용 효율성을 높이는 동시에 큰 사이즈 파일을 지원하도록 자동화 된 기능입니다.

오라클 11gR2 에서 1,4 그리고 16 의 allocation units (AU)의 가변 크기 extents 를 지원합니다. ASM 은 각 사이즈의 extents 에 미리 정의된 숫자를 사용합니다. 파일이 특정 임계값을 넘으면 다음 extent 사이즈가 사용됩니다.

파일의 extent 사이즈는 다음과 같습니다.

- 첫 20000 extent 는(0 – 19999) disk group AU 사이즈와 항상 같게 설정된다.
- 다음 20000 extent 까지(20000 – 39999) 4\*AU 사이즈와 같다.
- 다음 20000 extent 부터(40000+) 그 이상은 16\*AU 사이즈와 같게 설정된다.

적은 메모리와 파일을 설명하기 위해 필요한 적은 extent 포인터는 공유플에 extent map 관리에 필요하다. extent 사이즈는 파일 내와 파일 외부 모두 변화 한다.

다양한 사이즈의 extent 는 또한 100TB 의 Oracle databases ASM 을 사용하여 전개가 가능합니다.

그러나 많은 숫자의 비연속적인 작은 데이터 extent 가 할당과 해제 되어 지며 외부 단편화를 발생시키고 부가적이지 않은 연속적인 큰 extent 는 유효하다. 조각모음 작업은 모든 rebalance 작업의 일부로 통합됩니다. 그 결과 DBA 는 항상 rebalance 작업을 실행하여 디스크 그룹을 조각모음 해야 합니다.

그럼에도 불구하고 원하는 사이즈가 유효하지 않다면 ASM 는 자동으로 extent 배당에 대해 조각모음을 수행하기 때문에 이것은 매우 드물게 발생한다. 이것은 할당된 몇몇 긴 작업을 잠재적으로 만듭니다.

### ASM 제한 값

- 63 disk group
- 10,000 ASM disks
- 4 petabytes per ASM disk (2 TB if not on Oracle Exadata)
- 40 exabytes of storage (20 PB if not on Oracle Exadata)
- 1 million files per disk group
- Maximum file size:
  - External redundancy: 140 PB
  - Normal redundancy: 23 PB
  - High redundancy: 15 PB

AU의 기본 사이즈는 1MB입니다. 대부분의 경우 이 사이즈는 적절하지만 10TB가 넘는 대용량 데이터베이스를 생각하면 오브젝트는 1MB보다 큰 사이즈로 설정되었을 가능성이 높으며, AU 사이즈를 증가시켜 AU의 수를 줄이는 것이 바람직합니다. 11g에서는 데이터 그룹 생성 과정에서 `au_size`를 설정함으로써 간단하게 작업 할 수 있습니다.

```
CREATE DISKGROUP data6
EXTERNAL REDUNDANCY DISK '/dev/raw/raw413' ATTRIBUTE 'AU_SIZE' = '2M';
```

AU 사이즈는 1M, 2M, 4M, 8M, 16M, 32M, 64M 중 하나 이거나 바이트 단위로 절대값을 설정할 수도 있습니다.

```
CREATE DISKGROUP data6
EXTERNAL REDUNDANCY DISK '/dev/raw/raw413' ATTRIBUTE 'AU_SIZE' = '2097152';
```

디스크 그룹이 생성된 후에는 아래의 쿼리로 AU 사이즈를 점검 합니다.

```
SQL> select name, allocation_unit_size
2 from v$asm_diskgroup;
```

NAME	ALLOCATION_UNIT_SIZE
DATA	1048576
FLASH_BACK	1048576
OCR_VOTE	1048576
TEST	2097152

## 5. SYSASM 권한

ASM 인스턴스를 관리 할 수 있는 SYSASM role의 사용으로 DBA와 스토리지 관리자 사이에 중복을 피할 수 있게 되었습니다.

ASM 관리자 작업을 실행 할 수 있는 목적으로 만들어진 특별한 SYSASM 권한을 가진다. SYSASM권리는 데이터베이스 관리자로부터 ASM 관리를 분기시키고 SYSDBA 권한은 보안을 향상시킨다. 새로운 OS-privileged group인 OSASM group은 ASM을 위해 독점적으로 사용된다. 이 그룹의 멤버는 OS 인증을 사용하여 SYSASM으로 접속 할 수 있으며, ASM에 모든 접근이 가능합니다.

또한 새로운 SYSASM 유저를 생성하기 위해 ASM 인스턴스로 부터 CREATE USER와 GRANT SYSASM SQL 문을 조합하여 사용 할 수 있다. 이것은 로컬 혹은 원격 ASM 관리에 유리 할 수 있습니다. 이 명령은 각 ASM 인스턴스의 password file을 업데이트 합니다. 유사하게 REVOKE 명령을 통해 유저로부터 SYSASM 역할을 취소 하고, DROP USER 명령으로 password file로 부터 유저를 drop 합니다.

v\$pwfile\_users 뷰는 SYSASM 권한으로 접속하면 (TRUE)를 아니면 (FALSE)를 나타내는, SYSASM으로 불리는 새로운 컬럼을 포함합니다.

USERNAME	SYSDB	SYSOP	SYSAS
SYS	TRUE	TRUE	TRUE
ASMSNMP	TRUE	FALSE	FALSE

```

SQL> CONNECT / AS SYSASM
SQL> CREATE USER username IDENTIFIED by passwd;
SQL> GRANT SYSASM TO username;
SQL> CONNECT username / passwd AS SYSASM;
SQL> DROP USER username;

```

SYSDBA 권한은 ASM 인스턴스에서 제한적입니다.

## 6. ASM 디스크 그룹 호환성

각 디스크 그룹의 호환성은 별도로 관리 가능합니다.

- ASM 호환성은 디스크 구조의 **ASM metadata** 로 결정
- RDBMS 호환성은 고객 클라이언트 레벨의 최소값으로 결정
- ADVM 호환성은 **Oracle ASM Dynamic Volume Manager** 를 사용할 수 있을 경우 결정

각 디스크 그룹의 호환성은 독립적으로 제어됩니다. 이것은 **Oracle Database 10g** 와 **Oracle Database 11g** 의 디스크 그룹이 이 기종 환경에서 활성화 하는데 꼭 필요합니다. 3 가지 호환성은 각 **ASM** 디스크 그룹의 속성입니다.

- **RDBMS** 호환성은 디스크 그룹을 마운트하기 위해 인스턴스가 허락하는 **RDBMS** 인스턴스의 호환가능한 최소 버전입니다. 이 호환성은 **ASM** 과 데이터베이스 인스턴스 사이에 교환되는 메시지의 포맷을 결정합니다. **ASM** 인스턴스는 다른 호환성 설정으로 다른 **RDBMS** 클라이언트 실행 지원이 가능합니다. 데이터베이스 호환버전은 각 인스턴스 설정보다 크거나 해당 데이터베이스에 사용되는 모든 디스크 그룹의 **RDBMS** 호환성과 같아야 합니다. 데이터베이스 인스턴스는 전형적으로 **ASM** 인스턴스와 다른 오라클 홈에서 실행됩니다. 이 의미는 데이터베이스 인스턴스는 **ASM** 인스턴스가 아닌 다른 버전의 소프트웨어 버전에서 실행될 수 있습니다. 데이터베이스 인스턴스가 **ASM** 인스턴스에 먼저 연결되고 양쪽다 지원 가능한 가장 높은 버전을 협상합니다. 호환성 매개변수를 데이터베이스에 설정하고, 데이터베이스 버전과 **RDBMS** 호환성이 디스크 그룹의 설정이 데이터베이스 인스턴스가 주어진 디스크 그룹을 마운트 할수 있는지 여부를 결정합니다.

- **ASM** 호환성은 디스크에 **ASM** 메타데이터에 대한 데이터 구조의 형식의 제어로 지속적인 호환성 설정을 의미합니다. 디스크 그룹의 **ASM** 호환성 레벨은 같은 디스크 그룹의 **RDBMS** 호환성 레벨과 같거나 그 이상이어야 합니다. **ASM** 호환성은 **ASM** 메타데이터의 형식에 관계가 있습니다. 파일 콘텐츠의 형식은 데이터베이스 인스턴스에 의해 결정된다. 예를 들면 **RDBMS** 호환성이 10.1 로 설정 될 수 있는 반면 디스크 그룹의 **ASM** 호환성은 11.0 으로 설정할 수 있습니다. 이것은 디스크 그룹에 사용되는 소프트웨어 버전이 10.1 과 같거나 그 이상이라면, 소프트웨어 버전이 11.0 혹은 그 이상의 **ASM** 소프트웨어에 의해 디스크 그룹이 관리 되어 지는 것을 의미합니다.

- **ADVM** 호환성은 **Oracle ASM volume** 에 포함된 디스크 그룹을 통해 결정됩니다. 값은 11.2 이상 이어야 합니다. 이 속성을 설정 하기전의 **COMPATIBLE.ASM** 값은 반드시 11.2 이상이어야 합니다. 또한 **Oracle ADVM volume** 드라이버는 반드시 로드 되어야 합니다. 기본으로



**COMPATIBLE.ASM**의 속성은 설정하기 전에 비어있습니다.

디스크 그룹의 호환성은 프로토콜 메시지나 연속적인 디스크 구조의 변경에는 고급이 필요합니다. 그러나 향상된 디스크 그룹 호환성은 돌이킬 수 없습니다. 만약 향상된 디스크 그룹 호환성을 설정하였다면 이전 설정 값으로 변경할 수 없습니다. 이전 값으로 되돌리려면 예전 호환성 속성 설정으로 새로운 디스크 그룹을 생성하고 새로운 디스크 그룹에 이전 디스크 그룹의 데이터 파일을 복원합니다.

**CREATE DISKGROUP** 명령어와 **ALTER DISKGROUP** 명령어를 통해 디스크 그룹 호환성을 설정할 수 있습니다.

디스크 그룹 호환성에 덧붙여 말하면, **COMPATIBLE** 파라미터(database compatible version) 활성화 되어 특성을 결정합니다. 그것은 데이터베이스나 **ASM** 인스턴스의 **INSTANCE\_TYPE** 파라미터에 따라 적용됩니다. 예를 들면 10.1로 설정하면 Oracle Database 11g에서 소개한 모든 새로운 기능(disk online/offline, variable extents 등등)의 사용이 배제될 것 입니다.

```
SQL> select compatibility, database_compatibility
2  from v$asm_diskgroup
3  where name = 'DATA';
```

COMPATIBILITY	DATABASE_COMPATIBILITY
11.2.0.0.0	10.1.0.0.0

위에서 볼 수 있듯, **ASM Compatibility**는 11.2.0.0.0 설정되어 있습니다. 이는 디스크 그룹이 11.2 버전의 **ASM** 구조까지 지원할 수 있음을 의미합니다. 또 **RDBMS Compatibility**(위의 **DATABASE\_COMPATIBILITY** 컬럼)는 10.1로 설정되어 있습니다. 이는 **ASM** 디스크그룹 **DATA**는 10.1 버전 이후의 모든 **RDBMS**를 지원함을 의미합니다.

**ASM Compatibility** 속성을 설정하는 방법이 아래와 같습니다.

```
ALTER DISKGROUP data SET ATTRIBUTE 'COMPATIBLE.ASM'='11.2';
```

**ASM Compatibility**는 11.2로 설정되어 있지만 **RDBMS Compatibility**는 여전히 10.1로 설정되어 있습니다. 이 값을 11.2로 바꾸어 주는 방법은 아래와 같습니다.

```
ALTER DISKGROUP data SET ATTRIBUTE 'COMPATIBLE.RDBMS'='11.2';
```

한 가지 중요한 사실은 호환성은 전체 **ASM** 인스턴스가 아닌 개별 디스크그룹 단위로 설정됩니다. 디스크 그룹이 사용하는 버전에 따라 속성을 적절하게 설정하고 버전 간의 호환성 보장에 수반되는 오버헤드를 절감할 수 있습니다

## 7. ASM Disk Group Attributes

이름	소유권	값	설명
au_size	C	1 2 4 8 16 32 64MB	디스크 그룹의 AU 사이즈
compatible.rdbms	AC	Valid database version	DB와 ASM 사이의 메시지 교환 형식
compatible.asm	AC	Valid ASM instance	디스크의 ASM 메타 데이터 구조 형식

		version	
compatible.advm	AC	Valid database version	디스크 그룹이 Oracle ASM volume 에 포함 할 수 있는지 결정
access_control.enabled	A	TRUE   FALSE	디스크 그룹을 위한 Oracle ASM File Access Control 활성화를 결정한다
access_control.umask	A	{0 2 6}{0 2 6}{0 2 6}	새로 생성된 파일의 기본 권한 모드를 설정한다.
sector_size	C	'512'   '4096'   '4K'	디스크 그룹의 디스크에 확실한 sector 사이즈를 지정된 값으로 만든다.
disk_repair_time	AC	0 M to 2 <sup>32</sup> D	디스크를 제거 하기 전 OFFLINE 시간
template.tname.redundancy	A	UNPROTECT MIRROR HIGH	지정된 template 의 redundancy
template.tname.stripe	A	COARSE FINE	지정된 template 의 striping 속성

A: ALTER

C: CREATE

create 나 alter 로 ASM 디스크 그룹을 만들 때, CREATE DISKGROUP 이나 ALTER DISKGROUP 명령어의 새 ATTRIBUTE 절을 사용하여 소성을 변경 할 수 있습니다.

- ASM 은 디스크 그룹을 생성할 때 지정하여 다른 AU 사이즈의 사용을 가능하게 합니다. AU 사이즈는 1 MB, 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, or 64 MB 로 할 수 있습니다.
- RDBMS 호환성: "ASM Disk Group Compatibility" 에 더 많은 정보가 있습니다.
- ASM 호환성: "ASM Disk Group Compatibility" 에 더 많은 정보가 있습니다.
- DISK\_REPAIR\_TIME 을 minutes (M), hours (H), or days (D)로 지정할 수 있습니다.
- 이 값을 생략하면, 기본 값은 H 입니다. 속성을 생략하면 기본값은 3.6H 입니다. ALTER DISKGROUP 구문으로 속성을 무효화 합니다.

모든 속성을 SQL Language Reference 를 참고하여 전체 속성을 찾습니다.

정의된 디스크 그룹 각각에 V\$ASM\_ATTRIBUTE 고정 뷰를 사용하여 모든 정의된 속성을 볼 수 있습니다.

## 8. Enhanced Disk Group Checks

디스크 그룹 검사 구문이 간단해 졌다

- check 옵션을 FILE, DISK 그리고 ALL 로 대체

부가적인 검사 수행

- Alias
- Directories

```
ALTER DISKGROUP data CHECK;
```

---

CHECK 디스크 그룹 명령어는 기본으로 모든 메타 데이터 디렉토리에 간소화 됩니다. CHECK 명령어는 ASM 디스크 그룹 메타 데이터의 내부 일관성을 확인 합니다.

ASM 오류를 요약해 보여주고 alert log 에 감지된 오류를 자세히 기록한다. 이전 릴리즈에서 ALL, DISK, DISKS IN FAILGROUP, 그리고 FILE 로 이 절을 명시합니다. 이 절은 더 이상 필요 하지 않게 되었습니다. 현재 릴리즈에서 CHECK 키워드는 다음 명령을 실행합니다.

- 디스크의 일관성 checks 이전 버전의 CHECK DISK 와 CHECK DISK IN FAILGROUP 와 동 등)
- 모든 파일의 extent map 과 일관성을 위해 할당된 테이블의 Cross-checks (이전 릴리즈의 CHECK FILE 과 동등)
- 메타 데이터 디렉토리와 파일 디렉토리의 알리아스가 올바르게 링크 되었는지 check
- 디렉토리 트리 알리아스가 올바르게 링크 되었는지 check
- ASM 메타 데이터 디렉토리가 할당된 블록에 연결 할 수 없는지 check

REPAIR | NOREPAIR 절은 일관성 check 중 발견된 에러의 복구를 시도 할지 안 할지 ASM 에 설정합니다. 기본값은 REPAIR 입니다. 어떤 불일치성을 감시하고 싶을 때 NOREPAIR 로 설정하면 유용합니다. 그러나 그 문제를 해결하기 위해 어떤 자동 동작도 하지 않습니다.

## 9. Restricted Mount Disk Group for Fast Rebalance

- 하나의 디스크 그룹에는 하나의 싱글 인스턴스만 마운트 되어집니다.
- 데이터베이스 클라이언트가 없거나 다른 ASM 인스턴스는 접근 할 수 있습니다.
- locking overhead 없이 rebalance 진행 할 수 있습니다.

- ① ALTER DISKGROUP data DISMOUNT;
- ② ALTER DISKGROUP data MOUNT RESTRICT;
- ③ Maintenance task: Add/Remove disks ...
- ④ ALTER DISKGROUP data DISMOUNT;
- ⑤ ALTER DISKGROUP data MOUNT;

Oracle Datanase 11g 에서 디스크 그룹이 마운트 하기 위한 새로운 마운트 모드는 RESTRICTED 로 불립니다. 디스크 그룹이 RESTRICTED mode 로 마운트 되면, 클라이언트는 디스크그룹에 파일에 접근 할 수 없습니다. ASM 인스턴스가 클라이언트가 아닌 것을 알았을 때, 인스턴스는 클라이언트에 메시지의 locking/unlocking extent maps 시도를 하지 않아 rebalance 작업의 성능이 향상됩니다.

디스크 그룹이 RESTRICTED mode 로 마운트 되면 하나의 노드에만 독점적입니다.

RESTRICTED mode 는 내부적인 상호작용 없이 ASM 인스턴스의 디스크그룹의 모든 유지작업의 수행을 허락한다.

유지작업이 끝날 때 명시적으로 디스크그룹을 dismount 시키고, normal mode 로 remount 합니다.

ALTER DISKGROUP *diskgroupname* MOUNT 명령어는 RESTRICTED mode 에서 디스크 그룹을 마운트 하기위해 ASM 을 활성화를 확장합니다.

ASM 인스턴스를 시작하기 위해 **RESTRICTED** 옵션을 사용하면 **ASM\_DISKGROUPS** 파라미터에 정의 되어진 모든 디스크 그룹은 **RESTRICTED mode** 로 마운트 되어집니다.

## 10. Mount Force Disk Group

기본적으로 **MOUNT** 는 **NO FORCE** 입니다.

- 모든 디스크들은 유효해야 합니다.

**FORCE** 인 **MOUNT**:

- **Offline** 으로 사용할 수 없는 디스크가 퀴럼으로 존재하는 경우
- 모든 디스크 그룹이 실패한 경우

```
ALTER DISKROUP data MOUNT [FORCE|NOFORCE];
```

디스크 그룹이 불완전하게 마운트 되었을 때 **ASM** 에서의 행동입니다.

오라클 10g 에서, 디스크 그룹을 마운트 하기에 **failure groups** 는 충분했습니다. 심지어 없거나 손상된 **failure groups** 도 마운트 작업이 성공했습니다. 이 행동은 잠재적으로 자동으로 **ASM** 디스크를 드랍하고 수리가 끝난 후 다시 추가된 것을 요청하고, 이렇게 하여 긴 **rebalance** 작업을 초래합니다.

오라클 11 에서, 손상된 디스크그룹을 마운트 할 때 **FORCE** 옵션을 서술하지 않으면 이러한 명령은 실패합니다. (**ASM\_DISKSTRING** 이 올바르지 않을 때)에러가 맞다면 다시 마운트를 시도 할 수 있습니다.

그러나 하나 혹은 그 이상의 유효하지 않은 오프라인 디스크들을 가질 때 **FORCE** 옵션으로 디스크 그룹이 마운트 됩니다. 이 디바이스를 복구하기 위한 **DISK\_REPAIR\_TIME** 끝나기 전에 수정 작업을 해야 합니다. 이 디바이스의 실패는 디스크 그룹으로부터 디스크가 추방당하고 디스크 그룹의 모든 파일의 **redundancy** 를 복구하기 위한 고 비용의 **rebalancing** 의 결과를 초래합니다. 또한 **MOUNT FORCE** 결과 하나 혹은 그 이상의 디바이스가 오프라인 되었다면 몇 혹은 모든 파일이 올바르게 보호되지 않아 **rebalance** 를 통해 디스크 그룹에 **redundancey** 가 복구 됩니다.

그러므로 사용 불가능한 디스크 그룹에 속한 몇몇 디스크들은 **MOUNT** 의 **FORCE** 로 사용 가능합니다. **ASM** 이 찾은 충분한 디스크가 일정 숫자가 되면 디스크 그룹 마운트가 성공합니다. 아무것도 정의된 것이 없다면 **NOFORCE** 가 기본 옵션이다. **NOFORCE** 모드는 모든 디스크가 성공적인 마운트를 위해 접근 가능한 디스크 그룹에 속해야 합니다.

**NOTE:** 필요하지 않은 에러가 발생했을 때 **FORCE** 옵션을 정의합니다. 그것은 클러스터에서 특별합니다. 만약 **ASM** 인스턴스가 디스크 그룹을 처음으로 마운트 하지 않는다면 디스크들은 위치적으로 접근하기 어렵다고 결정하고 **MOUNT FORCE** 는 에러와 함께 실패하지만 다른 인스턴스에서는 접근 가능합니다.

## 11. Forcing Disk Group Drop

디스크가 더 이상 존재하지 않는 경우, 다시 말해 디스크가 복구 불가능한 수준으로 손상되었을 때에는 어떻게 해야 할까요? 이 디스크그룹을 영구적으로 드롭 처리하고 디스크그룹을 재

생성하거나 다른 디스크그룹의 디스크를 추가해야 할 것입니다. 하지만 드롭 처리 해야 하는 디스크그룹은 현재 마운트되지 않은 상태입니다. 디스크 중 하나가 누락된 상태이기 때문에 마운트 작업 자체가 불가능합니다. 디스크그룹을 드롭 처리하려면 마운트가 필요하지만 디스크가 손상되었기 때문에 마운트가 불가능한, 이도저도 못하는 상태가 된 것입니다.

10g에서는 dd 커맨드를 이용하여 디스크의 헤더를 삭제하는 우회적 해결책이 사용되었습니다.

```
$ dd if=/dev/zero of=/dev/raw/raw513 bs=1024 count=4
```

위 명령을 실행하면 /dev/raw/raw513 디스크의 헤더에 "0"의 값이 입력되고 모든 정보가 삭제됩니다. 이 방법은 효과적이긴 하지만 디스크 헤더의 정보를 완전히 삭제한다는 문제가 뒤따릅니다.

11g에서는 더 간단한 해결책을 사용할 수 있습니다. drop 명령을 실행하면서 force 옵션을 사용하면 됩니다.

DROP DISKGROUP FORCE 는 ASM 인스턴스에 의해 마운트 되지 않은 디스크 그룹에 속한 디스크의 헤더에 마크합니다. 그러나 ASM 인스턴스는 같은 스토리지 서브 시스템을 사용하는 어떤 다른 ASM 인스턴스에 의해 사용되는 디스크 그룹인지 우선 결정합니다. 같은 클러스터 혹은 같은 노드의 디스크 그룹이고 그것이 사용되면 명령문은 실패합니다.

디스크 그룹이 다른 클러스터에 있으면 다른 클러스터의 인스턴스에 의해 마운트 된 디스크 그룹을 결정에 덧붙여 시스템을 체크합니다. 디스크 그룹이 다른 곳에 마운트 되면 명령문은 실패합니다. 그러나 이후의 점검은 같은 클러스터의 디스크의 최후 점검이 아닙니다. 그러므로 주의 절을 사용해야 합니다.

NOTE: FORCE 옵션과 함께 DROP DISKGROUP 커맨드를 실행하면 INCLUDING CONTENTS 절을 정의해야 합니다.

```
DROP DISKGROUP data7 FORCE INCLUDING CONTENTS;
```

이제 디스크가 마운트되지 않은 상태에서도 디스크그룹이 드롭 처리됩니다. 사용 가능한 디스크는 "FORMER"로 표시됩니다(이는 디스크가 다른 디스크그룹에 포함되어 있었음을 의미합니다.)

## 12. ASMCMD Extension in Oracle Database 11g Release 1

ASMCMD 는 ASM metadata backup 과 restore 기능을 포함하여 확장 되었다. 이것은 정확한 템플릿과 디렉토리 구조에 대한 별칭을 포함한 이미 존재하는 ASM 디스크 그룹을 재 생성하는 능력을 제공합니다. ASM 디스크 그룹을 잃어버렸다면 RMAN 을 사용해 잃어버린 파일의 복구가 가능합니다. 그러나 ASM 디스크 그룹과 필요한 유저 디렉토리 와 템플릿은 메뉴얼하게 재 생성해야 합니다.

다음은 ASM metadata backup and restore(AMBR) 작업의 2 가지 모드입니다.

- 백업 모드 - AMBR 분석하면 존재하는 디스크와 failure group 구성, 템플릿, 디렉토리 구조 별칭에 대한 정보를 수집하기 위한 ASM 고정 테이블과 뷰 입니다.
- 복구 모드 - AMBR 은 디스크 그룹과 metadata 의 재 구성을 위해 이전에 생성된 파일을 읽습니다. full, nodg 혹은 newdg 를 restore 하는 복구모드에서 AMBR 행동을 통제하

는 능력이 있습니다. 다른 3 개의 서브 모드 사이에서 디스크그룹 생성과 변경 등 원하는 특색을 포함합니다.

lsdisk 커맨드는 ASM 디스크 정보 리스트입니다. 이 커맨드는 2 개의 모드에서 실행 할 수 있습니다.

- 접속 모드 - ASMCMD 는 디스크 정보를 검색하기 위해 v\$ 와 gv\$ 뷰를 사용한다.
- 비접속 모드 - ASMCMD 은 발견을 제한하기 위해 ASM 디스크 스트링을 사용하여 디스크 정보를 검색하기 위해 디스크 헤더를 스캔 합니다. 접속 모드를 항상 처음으로 시도한다.

cp 명령어는 로컬 인스턴스의 ASM 디스크 그룹과 리모트 인스턴스 사이에 파일 copy 가 가능하게 합니다. 예를 들면 다음과 같습니다.

```
cp +DATA/happy/datafile/test01.270.721133271 +DATA/happy/test01.bak
```

예제 가운데, 기존의 파일을 로컬로 copy 할 수 있다. 그러나 리모트 ASM 디스크 그룹으로 파일을 복사하기 위해 연결 문자열을 명시해야 합니다. 복사된 파일의 형식은 Little-Endian 과 Big-Endian 시스템 사이를 이동합니다. 또한 cp 명령어로 ASM 파일을 OS 로 복사할 수 있습니다.

```
cp +DATA/happy/datafile/test01.270.721133271 /backup/data/test01.dbf
```

비슷하게 cp 명령어로 OS 에서 ASM 디렉토리로 복사 할 수 있습니다.

```
cp /backup/data/test01.dbf +data/happy
```

만약 로컬 ASM 인스턴스에서 리모트 ASM 인스턴스로 ASM 파일을 복사하길 원한다면 다음 구문을 사용 할 수 있습니다.

```
cp +DATA/happy/datafile/test01.270.721133271 \sys@sky .+ASM2: +DATA/sky/test01.dbf
```

note: 이 명령어의 각 구문에 대한 더 많은 정보는 *Oracle Database Storage Administrator's Guide* 에서 확인 할 수 있습니다.

## 13. ASMCMD Command Extensions

ASMCMD 확장 명령어는 관리 명령어를 포함합니다.

- ASM 인스턴스 (startup, shutdown, disk statistics)
- ASM 파일 접근 제어 (유저와 그룹 접근의 생성, 관리)
- Disks and disk groups (create, mount, dismount, backup, restore, rebalance, set attributes)
- ASM files (aliases 의 생성과 삭제, 디렉토리 변경, 파일 리스트)
- ASM file templates (생성, 변경, 목록, 제거)
- ASM 볼륨 (생성, 제거, 사이즈변경, 통계정보)

ASMCMD 확장 명령어

Oracle Database 11g Release 1 에서, ASMCMD 는 18 개의 기본 명령어를 가지고 있었습니다.

Oracle Database 11g Release 2 에서 ASM 인스턴스, 파일, 디스크 그룹, 파일 접근, 템플릿, 볼륨을 관리하는 스토리지 관리자를 가능하게 하는 63 개의 유닉스와 비슷한 명령어를 가집니다. ASMCMD 명령어는 기본적으로 ASM 운영을 수행하는 SQL 명령어를 포함 합니다. 목표는

---

SQL 명령으로 수행할 수 있는 모든 ASM 작업도 ASMCMD 명령을 수행할 수 있습니다.

ASMCMD 명령은 시스템 관리자 및 스토리지 관리자에게 익숙하도록 유닉스와 비슷한 구문을 사용합니다. 명령어 확장은 관리자가 ASM 인스턴스와 스토리지를 관리하기 위해 SQL 을 배우는 것을 방지합니다. 명령어에 대한 자세한 내용은 Oracle Database Storage Administrator's Guide 11g Release 2 에 있습니다.

### ASMCMD Extension 예제

①

```
ASMCMD> md_backup meta_backup_fail01 -G fail01
Disk group metadata to be backed up: FAIL01
Current alias directory path: HAPPY/DATAFILE
Current alias directory path: HAPPY
ASMCMD>
```

②

```
Unintentional disk group drop
```

③

```
ASMCMD> md_restore meta_backup_fail01 --full -G fail01
Current Diskgroup metadata being restored: FAIL01
Diskgroup FAIL01 created!
System template AUTOBACKUP modified!
System template ONLINELOG modified!
System template OCRBACKUP modified!
System template ASMPARAMETERBAKFILE modified!
System template ASM_STALE modified!
System template ASMPARAMETERFILE modified!
System template PARAMETERFILE modified!
System template OCRFILE modified!
System template CHANGETRACKING modified!
System template ARCHIVELOG modified!
System template DUMPSET modified!
System template TEMPFILE modified!
System template XTRANSPORT modified!
System template FLASHBACK modified!
System template DATAGUARDCONFIG modified!
System template DATAFILE modified!
System template CONTROLFILE modified!
System template BACKUPSET modified!
System template FLASHFILE modified!
Directory +FAIL01/HAPPY re-created!
```



```
Directory +FAIL01/HAPPY/DATAFILE re-created!  
ASMCMD>
```

④

```
Restore disk group files by using RMAN.
```

이 예제는 `md_backup` 명령어를 사용하여 `ASM metadata` 를 어떻게 `backup` 하고, `md_restore` 명령어를 사용하여 어떻게 `data` 를 복구하는지 설명합니다.

첫번째 예제에 명령어에 `-G` 옵션이 서술되어 있습니다. 이것은 `backup` 에 필요한 디스크 그룹과 같은 `backup` 정보를 포함한 생성파일의 이름을 정의합니다.

두번째에서, `DATA` 디스크 그룹이 문제가 생겼을 경우를 가정합니다. 그 결과, 디스크 그룹을 제외합니다. 디스크 그룹에 포함된 데이터베이스 파일을 복구하기 전에 디스크 그룹 자체를 복구해야 합니다.

세번째, 디스크 그룹의 재생성 시키는 것은 `md_restore` 명령어를 사용하여 `metadata` 를 복구하는 것과 마찬가지로입니다. `step 1` 에서 생성된 `backup` 파일의 이름을 서술하는 것은 복구할 원하는 디스크 그룹의 이름과 같고, 더하여 원하는 복구 타입을 서술합니다. 이 예제에서는 디스크 그룹이 더 이상 존재하지 않아 전체 복구 했습니다.

디스크 그룹 재생성 후 `RMAN` 을 사용하여 데이터베이스 파일을 복구합니다.

## 14. ASM ACLs

- ASM file level permissions 설정
- Permissions 은 none(0), read(4), or read-write(6)
- Permissions 은 Linux 와 UNIX 운영체제에서만 사용 가능합니다.

ASM ACLs 는 ASM 파일의 선택적 보호를 제공합니다. ASM 접근 관리 리스트의 객체들은 안전하지 않지만 추가적인 파일 손상을 예방하기 위해 분리할 의무가 있습니다. ACLs 없이 SYSDBA 권한으로 모든 사용자가 그들은 제거하는 등 마운트된 디스크 그룹에 있는 ASM 파일에 액세스 할 수 있습니다.

Oracle ASM 파일 접근 제어를 설정하면, OSASM 을 위한 운영체제 그룹, ASM 을 위한 OSDBA, 데이터베이스 그룹을 위한 OSDBA 을 분리해서 생성해야 합니다. 데이터베이스를 위한 OSDBA 그룹은 같은 ASM 인스턴스를 사용하는 각 데이터베이스 인스턴스 마다 달라야 합니다.

각 ASM 파일은 DBUSER 에 의해 생성 됩니다. DBUSER 는 보통 데이터베이스 인스턴스의 소유자 입니다. ASM 파일은 DBUSER 에 의해 소유된 데이터베이스에 생성 됩니다. 운영체제의 사용자가 데이터베이스 소프트웨어 소유자의 ID 를 ASM 인스턴스에 있는 DBUSER 인지 확인 합니다. DBUSER 에 유효한 사용자 ID NUMBER 인지 운영체제에 의해 접근이 제한됩니다. 실행중인 데이터베이스 인스턴스의 운영체제 유저는 데이터베이스 인스턴스에 접근하여 디스크 그룹과 파일을 생성할 때 디스크 그룹에 자동적으로 추가됩니다.

각 DBUSER 는 접근 제어 목록을 생성할 수 있습니다.



ASM ACL은 유저 그룹을 포함한 DBUSER의 리스트이지만 password는 아닙니다. ASM은 OS 인증 메커니즘을 신뢰합니다. DBUSER는 ASM의 OSDBA 그룹의 멤버입니다.

**ASM 파일의 접근 제어 리스트에 필요사항:**

- 리눅스 혹은 유닉스 운영체제
- OS 레벨에서 분리된 역할
- 디스크 그룹 속성은 다음과 같이 설정
  - COMPATIBLE.ASM to 11.2 or higher
  - COMPATIBLE.RDBMS to 11.2 or higher
  - ACCESS\_CONTROL.ENABLED to TRUE
  - ACCESS\_CONTROL.UMASK to a mask value

접근 제어 리스트를 효과적으로 사용하기 위해 요구사항이 확실해야 합니다. ASM 접근 제어 리스트는 리눅스와 유닉스 운영체제에서만 사용 가능합니다. 반드시 job role을 분리해서 구성해야 합니다. 각 데이터베이스와 ASM 인스턴스는 다른 소유자여야 합니다.

디스크 그룹 속성은 반드시 설정 해야 합니다. COMPATIBLE.ASM과 COMPATIBLE.RDBMS는 11.2 이상으로 설정되어야 합니다. ACLs를 사용한 각 디스크 그룹은 ACCESS\_CONTROL.ENABLED을 TRUE로 설정하고, ACCESS\_CONTROL.UMASK 값을 설정합니다. ACCESS\_CONTROL.ENABLED은 ACCESS\_CONTROL.UMASK를 설정하기 전에 TRUE로 설정해야 합니다. umask 값은 소유자, 그룹, 타인에 읽기-쓰기의 모든 접근 권한을 제거합니다. 권한에서 6은 읽기-쓰기, 4는 읽기 0은 없음을 나타냅니다. 0의 umask 값은 제거되는 것이 없고, 2는 쓰기권한을 제거하고 6은 읽기-쓰기 권한을 제거합니다. 사용자, 그룹, 타인에게 권한에 연결된 값을 부여할 수 있습니다. 예를 들면, 026 umask는 소유자에게는 읽기-쓰기를, 그룹에게는 읽기를, 타인에게는 어떤 접근도 없는 640의 권한을 가진 파일로 변화시킵니다.

값을 설정하는 SQL의 사용은 다음과 같습니다.:

```
ALTER DISKGROUP DATA2 SET ATTRIBUTE 'access_control.enabled' = 'true';
ALTER DISKGROUP DATA2 SET ATTRIBUTE 'access_control.umask' = '026';
```

같은 ASMCMD 명령어는 다음과 같습니다.:

```
ASMCMD> setattr -d data2 access_control_enabled true
ASMCMD> setattr -d data2 access_control_umask 026
```

디스크 그룹의 속성은 ASMCA 틀에서도 설정 할 수 있습니다.

```
SQL> select name, value
2 from V$ASM_ATTRIBUTE
3 where group_number=4 and name like '%access%';
```

NAME	VALUE
access_control.enabled	true
access_control.umask	026

## 15. SQL 명령어 사용으로 ASM ACLs 관리

- SQL 명령어 사용으로 ASM ACL 관리:

```
ALTER DISKGROUP ADD USERGROUP ... WITH MEMBER
ALTER DISKGROUP DROP USERGROUP
ALTER DISKGROUP MODIFY USERGROUP ADD MEMBER
ALTER DISKGROUP MODIFY USERGROUP DROP MEMBER
ALTER DISKGROUP ADD USER
ALTER DISKGROUP DROP USER
ALTER DISKGROUP SET PERMISSION
ALTER DISKGROUP SET OWNERSHIP
```

- ACLs 에 대한 뷰 정보:

```
SELECT * FROM V$ASM_USER
SELECT * from V$ASM_USERGROUP
```

사용자 그룹을 생성할 때 ACL 명령어 사용:

```
SQL> ALTER DISKGROUP FAIL01 ADD USERGROUP 'test_grp1'
  2 WITH MEMBER 'oracle1','oracle2';

Diskgroup altered.
```

각 사용자는 반드시 존재 해야 합니다.

**SYSDBA** 나 **SYSASM** 권한을 가진 사용자는 그룹을 생성할 수 있습니다. 그룹 소유자나 **ASM** 관리자는 그룹을 변경, 삭제 할 수 있습니다. **ASM** 관리자만이 사용자를 추가, 삭제 할 수 있습니다. 파일의 소유자나 **ASM** 관리자 만이 파일의 소유권을 변경 할 수 있습니다.

파일의 사용자는 보통 데이터베이스 소유자이고, 디스크 그룹에 파일이 생성 되면 사용자는 자동적으로 추가 됩니다. 디스크 그룹에 사용자를 추가하면 거의 필요로 합니다. 데이터베이스 소유자가 아닌 **OS** 사용자는 디스크 그룹과 사용자 그룹에 추가 되어 합니다.

```
[root@goodus2:/root]# cat /etc/passwd
...중략...
oracle1:x:601:600::/home/oracle1:/bin/bash
oracle2:x:602:600::/home/oracle2:/bin/ksh
[root@goodus2:/root]#

SQL> SELECT group_number, os_name FROM V$ASM_USER;

GROUP_NUMBER OS_NAME
-----
4 grid
4 oracle1
4 oracle2
```

**SET PERMISSION** 절은 Oracle ASM 파일의 권한을 수정합니다. 참고로 읽기/쓰기 권한을 가진 파일에 일기-전용 권한을 설정하면 쓰기 권한을 회수 합니다.

```

SQL> ALTER DISKGROUP FAIL01 SET PERMISSION OWNER=read write, GROUP=read only,
  2  OTHER=none FOR FILE '+FAIL01/test01';

Diskgroup altered.

SQL> ALTER DISKGROUP fail01 SET OWNERSHIP OWNER='oracle1', GROUP='test_grp1'
  2  FOR FILE '+FAIL01/test01';

Diskgroup altered.

```

파일 소유자나 Oracle ASM 관리자는 파일에 대한 권한을 변경할 수 있습니다. 열린 파일의 권한은 변경할 수 없습니다. 열린 파일의 소유권도 변경할 수 없습니다.

```

SQL> ALTER DISKGROUP FAIL01 SET PERMISSION OWNER=read write, GROUP=read only,
  2  OTHER=none FOR FILE '+FAIL01/test01';
ALTER DISKGROUP FAIL01 SET PERMISSION OWNER=read write, GROUP=read only,
*
ERROR at line 1:
ORA-15032: not all alterations performed
ORA-15301: cannot change permission of the open file '+FAIL01/test01'

SQL> ALTER DISKGROUP fail01 SET OWNERSHIP OWNER='oracle1', GROUP='test_grp1'
  2  FOR FILE '+FAIL01/test01';
ALTER DISKGROUP fail01 SET OWNERSHIP OWNER='oracle1', GROUP='test_grp1'
*
ERROR at line 1:
ORA-15032: not all alterations performed
ORA-15301: cannot change group ownership of the open file '+FAIL01/test01'
ORA-15301: cannot change ownership of the open file '+FAIL01/test01'

```

## 16. ASMCMD 명령어 사용으로 ASM ACLs 관리

ASMCMD 명령어 사용으로 ASM ACLs 관리:

```

chgrp usergroup list_of_files
chmod mode list_of_files
chown user[:usergroup ] list_of_files
groups diskgroup user
grpmod { -a | -d } diskgroup usergroup user1 [user2]...
lsgrp [-Ha] [ -d diskgroup ] [ pattern_expr ]
lsusr [-Ha] [ -d diskgroup ] [ pattern_expr ]
mkgrp diskgroup usergroup [user1] [user2]...
mkusr diskgroup user
passwd user
rmgrp diskgroup usergroup
rmusr diskgroup user

```

ASMCMD 명령어는 스토리지 관리자에게 SQL 명령어와 같이 동일한 작업을 수행 가능하게 해주

지만 더 친숙한 형식입니다. ASMCMD 명령은 유닉스 명령어 형식입니다. ASMCMD 명령은 SQL 명령어과 같은 제한을 가지고 있습니다.

참고 사항:

- **chmod:** Oracle ASM 파일이나 파일의 리스트의 권한을 수정한다. 참고로 읽기/쓰기 권한을 가진 파일을 일기-전용으로 설정하면 쓰기 권한은 회수 됩니다. 파일 소유자나 Oracle ASM 관리자는 파일에 권한을 변경할 수 있습니다. 열린 파일의 권한은 변경할 수 없습니다.
- **chown:** 열린 파일에 대한 소유권은 변경할 수 없습니다.

```
ASMCMD> mkgrp fail01 test_grp2 oracle2
ASMCMD>
ASMCMD> lsgrp
DG_Name Grp_Name Owner
FAIL01 test_grp1 grid
FAIL01 test_grp2 grid
ASMCMD>
ASMCMD> lsusr
DG_Name User_Num OS_ID OS_Name
FAIL01 1 501 grid
FAIL01 4 500 oracle
FAIL01 2 601 oracle1
FAIL01 3 602 oracle2
ASMCMD>
ASMCMD> groups fail01 oracle1
test_grp1
ASMCMD> groups fail01 oracle2
test_grp1,test_grp2
ASMCMD> rmusr fail01 oracle1
ASMCMD> groups fail01 oracle1
test_grp1
ASMCMD> groups fail01 oracle2
test_grp1,test_grp2
ASMCMD>
ASMCMD> chown oracle2:test_grp2 '+FAIL01/test01'
```

## 17. ASM Intelligent Data Placement

이전 릴리즈에서 ASM은 ASM 파일 배치를 완벽하게 하기 위해 스트라이핑 기술을 사용한 교차된 ASM 디스크를 시도하였고, 하나의 동일한 오브젝트를 하나의 ASM 디스크에서 다뤘습니다. 하지만 물리적 디스크 영역은 디스크 스피들에서 더 멀었고, 바깥쪽 트랙은 안쪽 트랙 보다 더 큰 처리량을 호출했다. 그래서 교차된 디스크의 ASM 스트라이프 파일이 아닌 ASM 지능형 데이터 배치이나 그것도 수동으로 표시된 파일을 물리적인 디스크 영역에 대응하여 배치 됩니다. 그것은 파일의 우선 영역(HOT or COLD)에 근거한 ASM extents 할당입니다. 왜냐하면 지능형 데이터 배치는 스피닝 디스크 구조와 high-end 스토리지 array가 아닌 것을 타겟으로 하고 ASM은 특색을 이루는 물리적인 디스크에 대해 아무것도 알지 못하기 때문 입니다. HOT, COLD 두 개의 디스크 영역은 새로운 디스크 그룹을 생성하거나 기존 디스크 그룹에 디스크

다 추가할 때 인위적으로 식별 됩니다. 이러한 설정의 사용으로 빈번히 접근되는 데이터는 고 대역폭을 가지고 seek latency 를 줄이는 이러한 데이터의 배열로 트랙의 가장 바깥쪽에 위치 하도록 보장할 수 있습니다. 이것은 특히 많은 량의 데이터를 작은 작업 단위로 사용하는 응용프로그램과 관련이 있다. COMPATIBLE.ASM 과 COMPATIBLE.RDBMS 디스크 속성은 11.2 나 그 이상으로 설정하고 Intelligent Data Placement 를 사용해야 합니다.

참고: 디스크 영역은 접촉하는 트랙의 셋과 비슷한 성능의 특성을 가집니다. 바깥쪽 트랙은 길고 그러므로 안쪽 트랙보다 더 많은 sector 를 포함 합니다. 왜냐하면 회전 속도는 변하지 않고, 읽을 수 있는 sector 의 숫자는 바깥쪽 트랙이 보다 큼니다.

### ASM Intelligent Data Placement 예제

①

```
SQL> ALTER DISKGROUP datadg ADD TEMPLATE idptemp  
2 ATTRIBUTES (HOT MIRRORHOT);
```

②

```
SQL> CREATE TABLESPACE hottbs DATAFILE '+datadg(idptemp)';
```

③

```
SQL> ALTER DISKGROUP datadg ALTER FILE '+datadg.342.3'  
2 ATTRIBUTES (HOT MIRRORCOLD);
```

④

```
SQL> ALTER DISKGROUP datadg REBALANCE POWER 8;
```

첫 번째 문장은 datadg 디스크 그룹에 idptemp 라는 새로운 템플릿을 만듭니다. ASM Intelligent Data Placement 를 활성화를 설명하는 새로운 템플릿 속성을 확인 할 수 있습니다. (HOT | COLD 그리고 MIRRORHOT | MIRRORCOLD) 4 개의 속성을 묘사할 수 있습니다. 첫 두개는 기본 extents 에 위치를 정하기 위한 디스크 영역 지정을 위해 사용되고 마지막 2 개는 보조 extents 에 위치를 정하기 위한 디스크 영역 지정에 사용됩니다. 그래서 첫 번째 예제의 경우 idptemp 템플릿은 기본, 보조 extents 파일을 생성하기 위해 디스크 그룹 디스크의 hot 영역을 사용 할 수 있습니다.

두 번째 예제는, 이전에 생성된 idptemp 템플릿에 hottbs 라는 새로운 테이블 스페이스를 생성합니다. 이 ASM 파일의 생성은 datadg 디스크 그룹의 각 ASM 디스크의 hot 영역에 완전히 위치를 지정합니다.

세 번째 예제는 HOT 그리고 MIRRORCOLD 로 변경하여 기존 파일의 IDP 속성을 수정합니다. SQL 문장은 extent 이동을 시작하지 않습니다. 그러나 새로운 extents 는 새로운 지역에 따라서 할당됩니다.

그러나 마지막 예제는 새로운 지역에 정상적인 재분배 활동 이외에 +datadg.342.3 파일의 extents 이동을 시작하여 수동 재분배를 시작합니다.

참고: 희망하는 지역에 충분한 공간이 부족하면, ASM 다른 지역의 extents 를 할당하지만 그 지역 사이즈의 재 분배를 시작합니다.

## 18. ASM Intelligent Data Placement Best Practices

Read Frequency	Write Frequency	Primary region	Secondary region
High	High	HOT	MIRRORHOT
High	Low	HOT	MIRRORCOLD
Low	High	COLD	MIRRORHOT
Low	Low	COLD	MIRRORCOLD

ASM 현재 디스크 영역의 속성을 자동으로 결정할 수 없습니다. DBA 가 파일에 대한 활동에 따라 그것들을 결정하기 위해 필요합니다. 좋은 지표는 사용자 작업의 시작으로 파일이 접근 가능한지 아닌지 입니다.

예를 들면, 양쪽 archive log 와 backup 은 파일에 한번 쓰여지지만 지연 시간은 덜 중요해서 backup 은 off-peak 기간에 스케줄 될 것입니다. 다른 데이터베이스 활동과 안쪽 영역에 위치한 archive logs 는 데이터베이스 지연에 크게 불리할 지도 모릅니다. 또한, 몇몇 테이블스페이스 가 off-peak 기간에 update 만 된다면, 복제된 extents 는 안쪽 영역에 위치하게 될 것입니다. 위의 테이블은 기본과 보조영역에 정해진 파일의 접근 방식을 보여줍니다.

### ASM Intelligent Data Placement 확인

V\$ASM\_DISKGROUP 과 V\$ASM\_FILE 조회:

```
SELECT dg.name AS diskgroup, f.file_number, f.primary_region,
       f.mirror_region, f.hot_reads, f.hot_writes, f.cold_reads,
       f.cold_writes
FROM V$ASM_DISKGROUP dg, V$ASM_FILE f
WHERE dg.group_number = f.group_number and dg.name = 'dg_name';
```

V\$ASM\_DISKGROUP 과 V\$ASM\_TEMPLATE 조회:

```
SELECT dg.name AS diskgroup, t.name, t.stripe, t.redundancy,
       t.primary_region, t.mirror_region
FROM V$ASM_DISKGROUP dg, V$ASM_TEMPLATE t
WHERE dg.group_number = t.group_number and
       dg.name = 'dg_name'
ORDER BY t.name;
```

부적절한 파일 정책은 성능을 낮출 수 있습니다. 각 파일에 대한 접근 빈도수의 측정은 진단을 위한 도구로 중요합니다. 컬럼들은 ASM 고정 테이블이 추가되어 이 정보들을 제공합니다. 뷰 리스트의 아래에 포함된 컬럼은 Intelligent Data Placement 에 대한 정보를 제공합니다.

- V\$ASM\_DISKGROUP :
  - HOT\_USED\_MB : hot 영역에 사용되는 숫자의 megabytes
  - COLD\_USED\_MB : cold 영역에 사용되는 숫자의 megabytes
- V\$ASM\_FILE :

- 
- PRIMARY\_REGION : primary extents 할당에 사용되는 영역( HOT | COLD )
  - MIRROR\_REGION : mirrored extents 할당에 사용되는 영역( HOT | COLD )
  - V\$ASM\_TEMPLATE :
    - PRIMATY\_REGION : primary extents 할당에 사용되는 영역( HOT | COLD )
    - MIRROR\_REGION : mirrored extents 할당에 사용되는 영역( HOT | COLD )
  - V\$\_ASM\_DISK\_IOSTAT hot & cold 영역의 I/O 통계를 제공합니다.

V\$ASM\_FILE 의 컬럼은 다른 영역으로 이동시켜야 하는지 그 선택이 올바른지 확인하여 파일을 선택하는데 도움이 됩니다. V\$ASM\_DISK 와 V\$\_ASM\_DISK\_STAT 뷰는 V\$ASM\_FILE 과 같은 통계를 가지고 있지만 디스크 레벨에서 집계되었습니다.

지역의 변화가 실제 성능이 변경되었는지 V\$FILEMETRIC 과 V\$FILEMETRIC\_HISTORY 의 AVERAGE\_READ\_TIME, AVERAGE\_WRITE\_TIME, PHYSICAL\_READS 그리고

PHYSICAL\_WRITES 의 변경 전과 후, 혹은 AWR 리포트(특히 I/O 통계 부분의 File I/O 통계) 를 사용하여 확인합니다.