RMAN in praktijk

Deel 5: database copy naar ASM

In dit artikel over Oracle Recovery Manager gaan we het backup tool eens op een andere manier toepassen. Met behulp van RMAN gaan we een database migreren van een filesysteem naar Automatic Storage Management ofwel ASM. De uitdaging is dit te doen op een veilige manier met minimale downtime. Daarmee wordt ook echt minimale downtime bedoeld, want we brengen de database niet eerder down dan strikt noodzakelijk is en houden die periode zo kort mogelijk. Een extra moeilijkheidsgraad die we toevoegen is de standby database, die verbonden is en die we nog niet migreren naar ASM. We moeten de database dus houden zoals deze is en geen nieuwe incarnation creëren.

Migratie naar ASM

De migratie naar ASM vindt plaats in enkele stappen en de meeste daarvan kun je online uitvoeren. Het voorbeeld dat we gaan doorlopen is een standaard database verbonden met een standby database. De standaard database wordt naar ASM gemigreerd terwijl we de standby database ongemoeid laten.

De uitdaging is om de migratie uit te voeren met minimale downtime. De meeste stappen zijn online uit te voeren.

Indien de parameters *.db_create_file_dest en *.db_create_log_dest zijn gezet, dan worden alle data files, temp files en redolog files zonder naam aangemaakt in ASM.
Hierbij moet je wel rekening houden met de standby database, want als de parameter '*.standby_file_management' op auto staat zal de standby database ook zijn file in ASM willen plaatsen. En dat kunnen/willen we (nog) niet.

Om de werkwijze overzichtelijk te houden en controle over de downtime te hebben is er het volgende stappenplan:

- Zet de database parameters voor het gebruik van ASM
- Enable block change tracking
- Maak een backup copy van de database naar ASM
- Verplaats alle redolog files naar ASM
- Verplaats de tempfile naar ASM
- Maak een spfile aan in ASM
- Maak een incremental backup copy van de database naar ASM.
- Switch de database naar de kopie in ASM en start recover database.
- Kopieer de control file van het file system naar ASM
- Check de database en disable block change tracking.

Parameters voor het gebruik van ASM

In deze configuratie zijn drie parameters van belang, namelijk 'db_create_file_dest', 'db_create_log_dest' en 'standby_file_management'. De eerste twee zijn om aan te geven waar nieuwe files en redologs aangemaakt moeten worden. Indien een nieuw file aangemaakt wordt zonder naam en deze parameters staan gevuld met een diskgroup, dan worden deze automatisch aangemaakt in ASM. In een configuratie met een standby database worden nieuwe files op de standby automatisch aangemaakt op de standby als 'standby_file_management' op auto staat. Dat willen we niet, omdat op de standby geen ASM is geïnstalleerd.
Zodra de parameters zijn opgegeven, moeten we rekening houden met het vergroten van de database. Data files en redologs moeten met de hand op de standby worden aangemaakt.

```
SQL > alter system set standby_file_management=manual ;
 System altered.
 SQL > alter system set db_create_file_dest='+DG001' ;
 System altered.
 SQL > alter system set db_create_online_log_dest_1='+DG001' ;
 System altered.
```

De notatie +DG001 staat voor een diskgroup in ASM, een ASM

instance kan 1 of meer diskgroups bevatten. De fysieke opslag bevindt zich op de disks in de diskgroup.

Enable block change tracking.

Dit is een databaseparameter die gezet wordt, omdat deze niet in alle gevallen noodzakelijk is. Maar omdat deze in ons geval wel essentieel is, wordt deze apart genoemd. Met deze parameter gezet, worden alle gewijzigde blokken van de database sinds de laatste backup bijgehouden. Tussen de backup copy en het definitief omschakelen naar de datafiles in ASM kan enige tijd zitten. Een backup copy is een image copy van de databasefiles. Hier wordt geen compressie op uitgevoerd. Dit houdt in dat een recente backup weinig recovery tijd kent en geen restore tijd nodig heeft.
Om te voorkomen dat we een hele lange downtime krijgen, omdat we zo lang moeten recoveren, doen we op het laatste moment een incremental level 1 backup copy. Daarmee worden alle gewijzigde blokken sinds de eerste back-up copy gekopieerd. Alleen de laatste transacties moeten dan nog gerecovered worden. Daarmee zorgen we dat de downtime tot een absolute minimum beperkt blijft.

We doen op het laatst een incremental level 1 backup copy om te voorkomen dat we een te lange downtime krijgen.

```
SQL > alter database enable block change tracking ;
Database altered.
```

Backup copy van database naar ASM

Nu gaan we een image copy van de databasefiles in ASM maken, we doen een incremental backup. We zouden hier ook een level 1 kunnen opgeven, omdat deze automatisch een level 0 wordt als deze niet bestaat. We hoeven ons geen zorgen te maken over de duur dat de voorbereidingen ons kosten. Grote databasefiles nemen namelijk veel tijd in beslag, zolang de database beschikbaar blijft is dat geen probleem. Een level moet altijd aangegeven worden anders is een incremental niet mogelijk.
We gebruiken het volgende script:

```
connect target /
backup as copy incremental level 0 database format '+DG001' tag 'ORA_
ASM_MIGRATION';
```

Verplaatsen van redolog files naar ASM

De redologs kunnen verwijderd worden terwijl de database draait. Deze logs kunnen opnieuw worden aangemaakt in de diskgroup. Zolang ze maar niet in gebruik zijn. Dit geldt ook voor de standby redologs. Indien er meerdere members in een logfile groep zitten, dan moeten er altijd twee members overblijven. Je kunt er dan eerst nieuwe bijmaken en de oude verwijderen.
Standby; als je dit moet uitvoeren op een standby database dan kan een redolog file de status 'CLEARING' hebben. Dit kun je oplossen door 'alter database clear logfile group <number>' . Indien de status 'CLEARING_CURRENT' is dan moet op de primary kant een switch logfile gegeven worden.

Verplaatsen van temp file naar ASM

De tempfile kan op dezelfde manier als een redolog file naar ASM geplaatst worden. Eerst een nieuw file toevoegen in ASM en daarna de file op het filesystem verwijderen. Dit kan niet als de tempfile aangemaakt is als een 'bigfile' de tablespace kan dan alleen maar uit een enkele file bestaan.
In dat geval moet een nieuw temporary tablespace aangemaakt worden in ASM, dit moet dan de default temporary tablespace worden voor de database. Daarna kan de oude temporary tablespace worden verwijderd. Wat we hier doen is ook mogelijk onder RMAN, maar daarvoor moet de database down. Een van de doelen die we ons hier gesteld hebben, is de downtime zo kort mogelijk houden, dus wat we online kunnen doen, doen we ook online. Voor de volledigheid vermelden we toch even de syntax.

```
run{
switch tempfile 1 to '+DG001' ;
alter database open ;
 }
renamed tempfile 1 to +DG001 in control file

database opened
```

Als de database geopend wordt dan wordt de tempfile opnieuw aangemaakt.

Aanmaken van spfile in ASM

Er zijn twee soorten parameterfiles, een spfile die dynamisch is en een pfile die statisch is. Gedurende het proces worden beide gebruikt, pfile omdat je die kunt editen. De spfile is de file die we uiteindelijk gebruiken, omdat die in de ASM kan worden. geplaatst Dit is een lastig gedeelte, want een spfile kan wel alvast gemaakt worden met de nieuwe parameters. Het probleem zit hem in de controlfiles. De namen van de controlfiles worden gegenereerd op het moment dat deze gekopieerd worden. Omdat het path binnen de diskgroup bekend is, kan de controlfile entry wel alvast met een filenaam worden aange-

maakt. Bij het kopiëren van de controlfiles als Oracle Managed File wordt de naamgeving als alias gebruikt die naar het echte file wijst. Tijdens het kopiëren moet de database dan wel met de betreffende spfile worden gestart.
Eerst maken we een pfile van de huidig spfile, wijzigen daarin de benodigde parameters en bedenken de naamgeving voor de controlfiles.

```
SQL > create pfile='/tmp/initmvdb.ora' from spfile;

[initmvdb.ora] # ==> wijzigingen in het pfile voor het spfile wordt
gemaakt.
db_create_file_dest  ='+DG001'
db_create_online_log_dest1      ='+DG001'
db_create_online_log_dest2      ='+DG001'
standby_file_management         ='manual'
controlfile          ='+DG001/.../controlfile/controlfile1.
dbf','+DG001/.../controlfile/controlfile2.dbf'

SQL > create spfile='+DG001/<sid>/spfile<sid>.ora' from pfile='/tmp/
initmvdb.ora' ;

[init<sid>.ora] # init file die naar het spfile in asm wijst. Staat in
 default locatie
spfile='+DG001/<sid>/spfile<sid>.ora'
```

Incremental backup copy van database naar ASM

We zijn nu zover dat alles wat we kunnen voorbereiden ook daadwerkelijk is voorbereid. Nu gaan we nog eenmaal een incremental backup copy maken voor we de database stoppen. Dankzij deze handeling kunnen we later de recovery tijd zo kort mogelijk houden.
Het script dat we gaan uitvoeren is een incremental level 1, wat betekent dat we alle veranderingen sinds level 0 gaan uitvoeren. Let op, als er geen level 0 backup bestaat of gevonden wordt gaat RMAN een level 0 backup uitvoeren in plaats van een level 1. Theoretisch kan men met een script volstaan, als in het begin geen level 0 bestaat en dan een level 1 aangeeft, dan wordt er toch een level 0 gemaakt.
Dit heeft niet de voorkeur vanwege de onduidelijkheid, maar het is een mogelijkheid.

```
[level1BckCP.rmn]
connect target /
run{
 backup incremental level 1 for recover
 of copy with tag 'ORA_ASM_MIGRATION'
 database;
 recover copy of database with tag 'ORA_ASM_MIGRATION' ;
}
```

Switch de database naar de copy in ASM en start recover database (OFFLINE)

Nu komen we bij het gedeelte dat de database offline moet. Er zijn twee redenen. We kunnen de database niet recoveren als deze online is. Ook het control file kunnen we alleen maar kopiëren als de database offline is. Als eerste gaan we de database files in ASM gebruiken, daar voeren we een recovery op uit.

```
RMAN> startup force nomount pfile='/tmp/initmvdb.ora'
RMAN> alter database mount ;
database mounted
released channel: ORA_DISK_1
RMAN> switch database to copy ;
datafile 1 switched to datafile copy "+DG001/<sid>/datafile/
system.405.657122385"
datafile 2 switched to datafile copy "+DG001/<sid>/datafile/
undotbs1.403.657122407"
datafile 3 switched to datafile copy "+DG001/<sid>/datafile/
sysaux.404.657122399"
datafile 4 switched to datafile copy "+DG001/<sid>/datafile/
users.401.657122415"
RMAN> recover database ;
Starting recover at 11-JUN-08
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=321 devtype=DISK
starting media recovery
archive log thread 1 sequence 13 is already on disk as file
+DG001/<sid>/onlinelog/group_3.398.657123867
archive log thread 1 sequence 14 is already on disk as file
+DG001/<sid>/onlinelog/group_1.400.657124399
archive log thread 1 sequence 15 is already on disk as file
+DG001/<sid>/onlinelog/group_2.399.657124351
archive log filename=+DG001/<sid>/onlinelog/group_3.398.657123867
thread=1 sequence=13
archive log filename=+DG001/<sid>/onlinelog/group_1.400.657124399
thread=1 sequence=14
archive log filename=+DG001/<sid>/onlinelog/group_2.399.657124351
thread=1 sequence=15
media recovery complete, elapsed time: 00:00:04
Finished recover at 11-JUN-08
```

Copy de control file van het filesystem naar ASM

Dit is een lastig verhaal, in de manuals staat dat je hier voor backup en restore kunt gebruiken. Wat wel heel goed werkte was een restore commando van het echte control file naar ASM. Niet de current control file uit de backup set maar de daadwerkelijke file zelf. Alle andere methodes werkten alleen als je de database opende met een reset logs. Dat willen we juist niet, omdat we een standby database hebben die nog niet is gemigreerd naar ASM. De instance moet gestart zijn met de parameters voor de ASM configuratie, de namen van de control files worden gebruikt als alias.

```
RMAN> startup nomount pfile='/tmp/initmvdb.ora' from spfile;
RMAN> restore controlfile to '+DG001/<sid>/controlfile/
controlfile_<sid>.ctl'
2> from '/disk/oradata/<sid>/control01.ctl' ;
RMAN> startup force mount pfil'/disk00/oracle/admin/<sid>/pfile/
init<sid>_p.ora'
RMAN> alter database open ;
```

Check de database en disable block change tracking

Nu hebben we een database in ASM draaien. Voordat we het systeem vrijgeven, moeten we eerst controleren of er niets is overgeslagen of over het hoofd gezien.
We vragen alle locaties van databasefiles op van de database;

```
/*
file       : checkDBfiles.sql
creator    : Rick van Ek
date       : 12-06-2008
*/

select           name
from             v$datafile
union
select           name
from             V$tempfile
union
select            member
from             v$logfile
union
select           rtrim(value)
from             v$parameter
where            name in ('control_files','spfile')
order            by 1
/
```

Omdat we nu klaar zijn en geen image copies meer maken kan de 'block change tracking' uitgezet worden.

```
SQL> alter database disable block change tracking ;
```

Samenvatting

Doordat RMAN vanuit de instance werkt en onafhankelijk is van opslagmedium kun je eenvoudig een database in en uit ASM brengen/halen. De voorbereiding is tijdsonafhankelijk, dus daar kunnen geen knelpunten liggen. Voorbereidingstijd kan overbrugd worden door een incremental backup. Een incremental backup copy kun je optimaliseren door 'Block change tracking' aan te zetten. De downtime is minimaal, belangrijkste is daarbij het kopiëren van de controlfile. RMAN wordt in feite als een kopieerprogramma gebruikt.

Rick van Ek is Oracle Senior DBA bij Van Ek IT-Consultancy B.V.