

RMAN in praktijk

Deel 4: 'duplicate' database

In dit artikel over de Oracle Recovery Manager gaan we het hebben over het dupliceren van databases, ook wel 'clone' databases genoemd. In de praktijk kan het voorkomen dat je een kopie database nodig hebt. Dit kan nodig zijn voor verschillende toepassingen en/of onderzoeken. Als men ervaring heeft opgebouwd, kan men zelfs even een database dupliceren voor het onderzoeken en oplossen van problemen. Hetzelfde geldt voor proefmigraties, testen van updates etc. Dit kan zonder de productie te beïnvloeden in een identieke (actuele) omgeving en daar gaat het hier om.

Om een duplicaat database te maken moeten enkele zaken van te voren geregeld zijn. Dit zijn enkele eenvoudige stappen die, als het geregeld voorkomt, makkelijk in een script gezet kunnen worden.

Als eerste moet er een locatie worden aangemaakt voor de configuratie en de log- en trace files. Dit wordt in de \$ORACLE_BASE/admin/\$ORACLE_SID directory aangemaakt. Hierin wordt ook het parameter file geplaatst dat we nodig hebben. Aanpassingen aan het parameter file zijn afhankelijk van de vraag of de database een andere naam krijgt en de database files een andere locatie. In het eerste voorbeeld gaan we uit van een duplicaat database op hetzelfde systeem.

```
cd $ORACLE_BASE/admin ; mkdir test1 ; cd test1 ; mkdir adump bdump
cdump udump pfile

SQL> create pfile='/oracle/admin/test1/pfile/inittest1.ora' from spfile
;

File created.
```

Het aangemaakte parameter file moet worden aangepast, overal waar <sid> staat moet <sid> komen. Tevens worden twee parameters toegevoegd. Omdat we op hetzelfde systeem zitten moeten we voorkomen dat de files van de productie database worden overschreven. RMAN kent de locatie van de database en zal deze niet zonder meer overschrijven. Indien een locatie hetzelfde is zal een foutmelding worden gegenereerd.

```
[inittest1.ora]
db_file_name_convert=('sid','sid1')
log_file_name_convert=('sid','sid1')
...

ln -s /oracle/admin/test1/pfile/inittest1.ora $ORACLE_HOME/dbs/
inittest1.ora
```

De parameter *_file_name_convert zorgt dat de database files op een andere locatie worden opgeslagen. Je hoeft overigens niet de volledige path namen te geven. De betreffende parameters doen string replacements, je kan meerdere string paren opgeven. De paden voor de database- en redolog-files moeten van te voren zijn aangemaakt. Om de Oracle environment te kunnen zetten moet er een entry in /etc/oratab gemaakt worden, voor Windows omgevingen moeten de services aangemaakt worden met Oradim. Nu kan de instance worden gestart.

```
[/etc/oratab]
#
#
test:/oracle/product/10.2.0/db_2:N
test1:/oracle/product/10.2.0/db_2:N

[rickvek@centos5_00 bin]$ . oraenv
ORACLE_SID = [test] ? test1

[rickvek@centos5_00 dbs]$ orapwd password=xxxxxx file=orapwtest1
[rickvek@centos5_00 dbs]$ sqlplus / as sysdba
SQL > startup nomount pfile='/oracle/admin/test1/pfile/inittest1.ora'
[listener.ora]
LISTENER =
  (DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (PORT =
61521))))

SID_LIST_LISTENER =
  (SID_LIST = (SID_DESC = (SID_NAME = test)
                (SID_DESC = (SID_NAME = test1)))

[tnsnames.ora]
TEST = (DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP) (HOST = centos5_00) (PORT = 61521))
        (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = test)))
# dupliceren op eigen systeem
TEST1 = (DESCRIPTION =
         (ADDRESS = (PROTOCOL = TCP) (HOST = centos5_00) (PORT = 61521))
         (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = test1)))
# dupliceren op ander systeem
TEST2 = (DESCRIPTION =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = centos5_01)(PORT = 61521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = test2)))
```

Locaties voor de database files moeten toegankelijk zijn c.q. de paden moet aangemaakt zijn. Het RMAN proces kan niet de volledige paden aanmaken, dus dat is een onderdeel van de voorbereidingen. Let op: de listener moet voor de sid geconfigureerd zijn, of je moet een entry in je network configuratie opnemen voor de 'duplicate' database. In dit geval is de sid van de 'duplicate' database opgenomen in de network configuratie. Er is ook een configuratie opgenomen voor een duplicate database op een andere server.

Als voorbereiding hebben we dus de volgende stappen gedaan:

1. Paden voor configuratie aangemaakt;
2. Parameter file aangemaakt;
3. Toegang voor database files aangemaakt;
4. Environment gezet;
5. Password file aanmaken in dbs directory onder Oracle home. (database voor windows);
6. Network configuratie wordt aangemaakt;
7. Instance in nomount gestart.

Om de database te dupliceren op hetzelfde systeem gebruiken we het onderstaande script;

```
connect target sys/<pwd>
connect auxiliary sys/<pwd>@test1
run {
  allocate auxiliary channel aux1 device type disk ;
  allocate auxiliary channel aux2 device type disk ;
  allocate auxiliary channel aux3 device type disk ;
  allocate auxiliary channel aux4 device type disk ;
  'duplicate' target database to test1
  db_file_name_convert=('test','test1');
}
```

Het RMAN script maakt eerst een connectie naar de database en daarna een connectie naar de processen van de 'duplicate' database. De rest moet in een run block omdat dat de enige manier is om channels te definiëren voor de 'duplicate' database. In dit voorbeeld zijn meerdere 'channels' benut omdat we snel willen zijn en toch geen resources nemen van de productie database. De rest wordt allemaal afgehandeld door het 'duplicate database' commando. Bij het 'duplicate' commando kan men de parameter `db_filename_convert` opgeven, dit is hetzelfde als in het parameter file, maar dit kan alleen voor de database files en niet voor de logfiles. De voorkeur is dan toch het parameter file.

De volgende acties worden door RMAN uitgevoerd;

Het SCN, System Change Number, tot waar deze gerecovered moet worden, wordt bepaald, de nieuwe namen voor de paden van de 'duplicate' database worden gezet. Vervolgens worden de datafiles gerestored naar die locatie.

Een nieuw control file wordt aangemaakt met de nieuwe data-

base naam en de nieuwe redo log file locaties. Alle database files worden omgezet naar de nieuwe locatie. Nu kan de database worden gerecovered tot het aangegeven SCN. De database wordt opnieuw gestart en een nieuw control file wordt aangemaakt, de gerecoverde database files worden in de recovery catalog genoteerd. Op dat moment wordt ook herkend dat het tempfile niet aanwezig is. Deze wordt dan opnieuw aangemaakt. Waarna de database wordt geopend, open resetlogs, en nieuwe redolog files worden aangemaakt.

De omschrijving is heel summier, maar geeft wel aan wat er gebeurt, voor de details hoeft men het niet te doen omdat alles door scripts wordt afgewikkeld. Wel is het verstandig om te weten wat er gebeurt voor het geval er een fout optreedt. Dan kan worden beoordeeld of opnieuw moet worden gestart, of het proces kan doorlopen, of hoe men fouten kan oplossen. Een veel voorkomende fout is dat de restore niet goed loopt, meestal doordat men niet genoeg ruimte heeft gereserveerd of een foutieve locatie heeft aangegeven. Na correctie kan het proces worden voortgezet. RMAN herkent de files die reeds gedaan zijn en hun status. De files die correct zijn worden overgeslagen en daarna gaat het script weer verder. Een andere veel voorkomende fout is dat aan het eind de database gestopt en gestart wordt. We bedoelen hier de 'duplicate' database. Het kan gebeuren dat er parameters in de init.ora file staan die voorkomen dat de database wordt gestart. Of het parameter file zelf staat niet op een default locatie, waardoor deze niet wordt gevonden.

In dat geval kan je de database openen met resetlogs waardoor de redolog files worden aangemaakt.

Vervolgens moet je controleren of het temp file is aangemaakt. Indien deze er niet is kan je een nieuw temporary file aanmaken. Als de database nog niet gerecovered is, is het beter om het script opnieuw te starten. Laat de database het maar zelfstandig regelen.

Duplicate database naar een andere server

Voor de opdracht duplicate database hebben we meer voorbereiding nodig. We hebben een server nodig met daarop geïnstalleerd dezelfde versie van Oracle software. Daarnaast moeten we de listeners configureren voor de duplicate en de productie database. Als men op beide systemen de volledige network configuratie aanmaakt dan maakt het niet uit vanuit welk systeem het script met het duplicate commando wordt gestart. Het script kan universeel worden gemaakt door in beide connect descriptions de TNS, Transparent Network Substrate, namen te gebruiken. Omdat men op een ander systeem werkt kan men dezelfde filenamen en paden gebruiken. Het is zelfs mogelijk om dezelfde database naam te gebruiken. Hierbij zal RMAN in eerste instantie een foutmelding geven. In het duplicate commando moet dan de parameter 'nofilenamecheck' opgegeven worden.

De volgende stappen zijn ter voorbereiding uitgevoerd;

1. Oracle software installatie.
2. Toegang voor database files aangemaakt.
3. Paden voor configuratie aangemaakt.
4. Parameter file aangemaakt, kan een kopie van origineel zijn.
5. Environment gezet.
6. Password file aangemaakt.
7. Network configuratie aangemaakt.
8. Listener gestart.
9. Instance in nomount gestart.

```
connect target sys/<passwd>@test
connect auxiliary sys/<passwd>@test2
spool msglog to '/oracle/general/log/clonedb_2.log' ;
spool trace to '/oracle/general/log/clonedb_2.trc' ;
run {
  allocate auxiliary channel aux1 device type disk ;
  allocate auxiliary channel aux2 device type disk ;
  allocate auxiliary channel aux3 device type disk ;
  allocate auxiliary channel aux4 device type disk ;
  debug on;
  duplicate target database to test2
    db_file_name_convert=('test','test2');
}
```

Omdat de duplicate database op een andere server ligt kan men volstaan met de originele naam. Dit maakt de voorbereiding erg eenvoudig, aangezien de configuratie files makkelijk kunnen worden gekopieerd. Het commando voor duplicate database wordt dan 'duplicate target database nofilenamecheck;'. De laatste parameter is noodzakelijk omdat RMAN altijd zal proberen te voorkomen dat de files van de target database worden overschreven. Wel moet men er voor zorgen dat de backup toegankelijk is op het andere systeem. In de praktijk houdt dit in dat de betreffende backup files naar dezelfde locatie op de andere server wordt gekopieerd. Tenminste uitgaande van een backup op disk, anders moet de tape unit geconfigureerd worden op het andere systeem. Indien de locatie anders is moet voor het device worden aangegeven waar deze zijn backup files kan vinden.

Duplicate database met PIT

Het is mogelijk om de database te dupliceren met een time-stamp die in het verleden ligt. Het mechanisme hiervoor is hetzelfde als een 'Point In Time Recovery' welke bij de restore is behandeld. Als men in het logfile van de duplicate database kijkt valt op dat in het eerste memory script RMAN zelf al een PIT doet, maar dan tot het laatst mogelijke moment.

```
contents of Memory Script:
{
  set until scn 949519;
  ...
}
```

Nu kan dat ook in ons startscript worden gedaan, maar dan met een SCN zoals wij die willen. Nu moet het wel een geldige

scn zijn zodat de backup de juiste restore informatie bevat. Dit is eenvoudig te controleren door een restore preview script te maken, die geeft dan aan welke backup nodig is en of die beschikbaar is. De view `v$archived_log` geeft genoeg informatie om de initiële keuze te maken.

```
SQL> @loginfo
```

RECID	STAMP	THREAD#	SEQUENCE#	FIRST_CHANGE#	NEXT_CHANGE#
1	690905038	1	8	660728	681465
2	690914886	1	9	681465	688247
--- cut					
11	691783514	1	18	918403	947160
12	691786653	1	19	947160	949519

Hieronder een voorbeeld voor het controleren van de juiste SCN door middel van een restore preview. Let wel: in het voorbeeld is slechts een beperkte set van backups beschikbaar. Hoe completer de cyclus is, des te makkelijker is het om terug in tijd te gaan met de duplicate database.

```
[rickvek@centos5_00 rman]$ rman cmdfile test_pit.rmn
Recovery Manager: Release 10.2.0.1.0 - Production on Fri Jul 10
00:24:02 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.
RMAN>
RMAN> connect target *
2> run{
3> set until scn 949427 ;
4> restore database preview ;
5> }
6>
connected to target database: TEST (DBID=1988620981)
executing command: SET until clause
using target database control file instead of recovery catalog
Starting restore at 10-JUL-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=146 devtype=DISK

List of Backup Sets
=====
BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
20 Full 19.70M DISK 00:02:19 09-JUL-09
BP Key: 20 Status: AVAILABLE Compressed: YES Tag:
TAG20090709T185257
...

rman cmdfile=dupdbpit.rmn using 949427

[dupdbpit.rmn]
connect target sys/<passwd>@test
connect auxiliary sys/<passwd>@test2
spool msglog to '/oracle/general/log/clonedb_2.log' ;
spool trace to '/oracle/general/log/clonedb_2.trc' ;
run {
  set until scn &1 ;
  allocate auxiliary channel aux1 device type disk ;
  duplicate target database ;
}
```

Als de regel 'set until scn 949427;' aan het script wordt toegevoegd dan kunnen we daarmee een database dupliceren. Het recovery proces stopt dan bij het aangegeven SCN, dat is te controleren door 'current_scn' uit de view `v$database` op te vragen

direct na het openen van de database. In het bovenstaande voorbeeld is een script gebruikt met de aanroep van een parameter. De variabele &I wordt gevuld met de waarde die als eerste achter de aanroep parameter 'using' wordt genoemd.

Duplicate database for standby

Deze vorm is heel bijzonder, je kan namelijk rman een standby database klaar laten zetten zonder dat de primary, productie, database wordt beïnvloed. Als we de configuratie van de listeners voor redolog transport even buiten beschouwing laten dan lijkt het veel op het dupliceren van de database op een ander systeem. Per slot van rekening is de standby database niets anders dan een physical database die constant staat te recoveren. De enige afwijking zijn de controlfiles, daarom moet een backup van een controlfile gemaakt worden als zijnde voor een standby.

```
connect target /
# connect catalog rman/recover@catalog
spool msglog to '/opt/oracle/app/oracle/admin/general/log/Bu4Stndb.log'
append ;
configure controlfile autobackup off ;
run {
  backup as compressed backupset database ;
  backup as compressed backupset current controlfile for standby ;
}

configure controlfile autobackup on ;
```

Het formaat van de controlfile is daarmee gelijk geschikt om een standby database te creëren. Nu moeten we de database dupliceren zoals we dat eerder ook hebben gedaan maar dan met de zojuist gemaakte backup van het controlfile.

```
connect target sys/<pwd>@MyDB
connect auxiliary /
# connect catalog rman/<pwd>@catalog
spool msglog to '/opt/oracle/app/oracle/admin/general/log/CreStndb.log'
append
run {
  allocate auxiliary channel aux1 device type disk ;
  allocate auxiliary channel aux2 device type disk ;
  allocate auxiliary channel aux3 device type disk ;
  allocate auxiliary channel aux4 device type disk ;
  duplicate target database for standby nofilenamecheck ;
}
```

Na het uitvoeren van het bovenstaande script is de database gestart als een managed standby database met auto recovering aan. Het enige wat nu nog ontbreekt is een transport van redo-logfiles van de primary database naar de standby database. Het handigste is om aparte listeners op beide databases te configureren voor het transport deze op te nemen in een tnsnames.ora configuratie file. Als men dat van te voren regelt kan de database zodra deze in recovery mode komt alle files recoveren zonder dat eerst de ontbrekende archived redo-log files opgehaald moeten worden over het netwerk. Een voorbeeld van de benodigde database parameters volgt hieronder

```
# parameter on standby
alter system set log_archive_dest_1='service=MyDB_FAL1 valid_
```

```
for=(online_logfile,primary_role) db_unique_name=MyDB1 lgwr async
delay=5 reopen=60 max_connections=3' scope=memory
/
alter system set fal_client='MYDB_FAL2' scope=memory ;
alter system set fal_server='MYDB_FAL1' scope=memory ;

#parameters voor de primary database.
alter system set log_archive_dest_2='service=MyDB_FAL2 valid_
for=(online_logfile, primary_role) db_unique_name=MyDB2 lgwr async
delay=5 reopen=60 max_connections=3' scope=memory
/

alter system set log_archive_dest_10='location="USE_DB_RECOVERY_FILE_
DEST", valid_for=(ALL_LOGFILES,ALL_ROLES)'
/

[tnsnames.ora] (een entry voor de primary en een voor de standby.)
MyDB_FAL1 = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = OraLinEnt01i) (PORT = 61221))
  (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = MyDB)))

MyDB_FAL2 = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = OraLinEnt02i) (PORT = 61221))
  (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = MyDB)))

[listener.ora] (zelfde entry voor standby en primary.)
LSNR_MyDB_FAL = (DESCRIPTION_LIST =
  (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (PORT = 61221))))
```

Bij het voorbeeld hierboven behoort een listener configuratie waarbij op iedere server een LSNR_MyDB_FAL wordt gestart. FAL staat voor Fetch Archive Log, dit proces zorgt voor het transport van de redo-logs. De complexiteit zit in het correct configureren van het transport en niet in het opzetten van de database.

Samengevat

Bij het uitvoeren van een 'duplicate' op hetzelfde systeem ligt het knelpunt bij de veranderingen, die men in het init.ora file uitvoert. Database file locaties of ruimtegebrek kunnen als spelbreker optreden. Goede voorbereiding is dan ook essentieel. Bij het uitvoeren van een 'duplicate' op een andere server ligt het knelpunt bij de network configuratie en de toegankelijkheid van de backup. Indien de database file locaties hetzelfde blijven, moet de parameter 'nofilenamecheck' worden gebruikt. Omdat er disk channels voor het kopiëren van de datafiles geconfigureerd moeten worden, moet alles in een run block worden gezet.

Het is eenvoudig om een 'Point In Time' aan te geven, de 'duplicate' database wordt dan tot dat tijdstip of die transactie gerecovered. Indien een duplicate stopt i.v.m. een foutmelding is het proces eenvoudig te herstarten na het oplossen van de foutmelding. Dat wat al correct is afgehandeld wordt niet nogmaals gedaan. Het opzetten van een standby is even complex als het dupliceren van een database naar een andere server. Wel heeft deze meer voorbereiding nodig, met name voor het redo-log transport.



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