

Redden wat er te redden valt?

Databases met corrupted redolog-file

Het relaas van een herstelactie van databases waarvan de redolog-files verwijderd zijn en de actieve redolog-file gecorrumpeerd is geraakt. De databases draaien niet in archivelogmode en er bleek al een half jaar geen goede back-up te zijn gemaakt. De belangrijkste vraag luidt natuurlijk hoe je deze databases kunt herstellen. Hoewel dit (hopelijk) geen dagelijkse activiteit is van een Oracle DBA, een beschrijving van de gevolgde werkwijze. Met tips en hints om de problemen het hoofd te bieden.

De uitgangssituatie betreft een Windows 2000-server met daarop vijf Oracle-databases (versie 9.2.0.1.0). Op deze server zijn drie schijven aanwezig (C:, D: en F:). De E:-drive is voor de dvd-speler, wat doet vermoeden dat de F:-schijf later is toegevoegd. Op de C:-schijf de Windows-eigen programmatuur, op de D:-schijf een directory met de Oracle 9.2.0.1.0-software en een oradata-directory van een viertal databases. Op de F:-schijf een oradata-directory met de vijfde database.

De D:-schijf zit erg vol, wat leidt tot performanceklachten. Een systeembeheerder doet een 'search' op *.log op de C:- en de D:-schijf en vervolgens een 'shift-delete'. En daarmee is het kwaad geschied. Drie van de vier databases op de D:-schijf crashen. Al snel blijkt dat behalve gewone log-files ook de redologs (REDO*.log) van de databases verdwenen zijn, en zelfs de actieve redolog onbruikbaar is volgens Oracle.

De eerste (terechte) gedachte is het terugzetten van de back-up. Helaas blijkt deze back-up niet goed gegaan te zijn. De laatste goede back-up blijkt meer dan een half jaar oud te zijn, en dat is te lang geleden, dat zou teveel dataverlies opleveren. Wat te doen?

Eerste diagnose en herstel poging

Een eerste controle levert op dat de redolog-files op de D:-schijf van de gecrashte databases er afwijkend uitzien. Twee van de drie logfiles (REDO01.log en REDO03.log) hebben de grootte 0 KB en worden door Oracle gezien als respectievelijk INACTIVE en UNUSED. De derde (REDO02.log) is 102400 KB, met als status INVALIDATED. Ter vergelijking: de database op

de F:-schijf heeft drie logfiles van 102401 KB groot. Ook de trace-file van de logwriter levert informatie:

```
ORA-00316: log 2 of thread 1, type 0 in header is not log file
ORA-00312: on-line log 2 thread 1: 'D:\oracle\oradata\dw\REDO02.log'
```

Conclusie: de actieve redologfile is ondanks de lock van Windows toch beschadigd geraakt.

De eerste herstel poging om de logfiles opnieuw aan te maken, gebeurt via het hercreëren van de controlfile. Dit levert niet het gewenste resultaat op. De situatie met de logfiles verandert niet en de database is nog steeds niet open te krijgen.

Het is mogelijk om met hidden parameters (waardoor Oracle corrupte redolog-files en corrupte undo-segmenten 'accepteert') in combinatie met Automatic Undo een database te herstellen. Hiervoor heb je wel de medewerking van Oracle nodig.

Warning

The following instructions should only be used under the explicit direction of Oracle Support. These steps should only be used when all other conventional means of recovering the database have failed. Please note that there is no guarantee that this method will succeed.

IF THE STEPS BELOW DO ALLOW YOU TO OPEN YOUR DATABASE THEN IT IS ESSENTIAL THAT THE DATABASE BE REBUILT AS IT IS NO LONGER SUPPORTED. FAILURE TO DO SO MAY LEAD TO DATA DICTIONARY INCONSISTENCIES, INTERNAL ERRORS AND CORRUPTIONS.

THE INSTRUCTIONS HERE ARE DESTRUCTIVE. YOU ARE STRONGLY ADVISED TO BACKUP THE DATABASE BEFORE PROCEEDING. IF YOU DO NOT DO THIS YOU MAY LOSE THE CHANCE TO TRY OTHER OPTIONS.

Er wordt dus een Service Request bij Oracle neergelegd met een beschrijving van de situatie.

Het definitieve herstel

Oracle reageert snel. Oracle heeft Note 283945.1 "TECH: Summary For Forcing The Database Open With `_ALLOW_RESETLOGS_CORRUPTION` with Automatic Undo Management" opgestuurd. Deze note is niet op Metalink te vinden, hij is dan ook via de mail gestuurd. De note klinkt wat dreigend (zie pag. 6, 'Warning').

De eerste database wordt onderzocht (in dit artikel EEN genoemd), na het maken van een koude back-up. De klant gebruikt spfiles, dus eerst dienen we een pfile te maken. Dat gaat als volgt:

```
SQL> startup mount
SQL> create pfile from spfile;
SQL> shutdown immediate
```

In deze `initEEN.ora` wordt het volgende veranderd:

```
##.UNDO_MANAGEMENT=AUTO
##.UNDO_TABLESPACE=UNDOTBS
##.UNDO_RETENTION=3600
UNDO_MANAGEMENT=MANUAL
_ALLOW_RESETLOGS_CORRUPTION = TRUE
_CORRUPTED_ROLLBACK_SEGMENTS = (_SYSSMU1$, _SYSSMU2$, _SYSSMU3$,
SYSSMU4$, _SYSSMU5$, _SYSSMU6$, _SYSSMU7$, _SYSSMU8$, _SYSSMU9$, _
SYSSMU10$)
```

Kortom, Automatic Undo uitzetten en tegen Oracle zeggen dat redo- en undo-corruptie is toegestaan. Die namen van de undo-segmenten zijn de default-namen, maar voor alle zekerheid is dat gecontroleerd aan de hand van de `alertEEN.log`. De spfile wordt hernoemd zodat Oracle de pfile gebruikt om de database te openen:

```
SQL> startup mount
```

Vervolgens controleren of alle parameters naar wens staan (met behulp van `show parameter corrupt` en `show parameter undo`) en dan kijken of alle datafiles ONLINE of SYSTEM zijn:

```
SQL> select name,file#,status from v$datafile where status not in
('ONLINE','SYSTEM');
```

Alle datafiles zijn ONLINE, de volgende stap is het uitvoeren van een incomplete recovery:

```
SQL> recover database until cancel;
```

direct gevolgd door CANCEL. Hierna proberen de database open te krijgen:

```
SQL> alter database open resetlogs;
```

Het resultaat hiervan is:

```
ORA-03113: End-of-file on communication channel. Disconnection forced.
```

Helaas, de instance crasht dus. Oracle heeft dit in note 283945.1 ook al voorzien als een mogelijkheid en voor een speciaal geval ook opgelost. Dit speciale geval is een ORA-00600 [2662]. Als deze foutmelding in de gegenereerde trace-file of in de `alert.log` voorkomt bij de crash, gloort er hoop. De ORA-00600 [2662] staat er in, en dit is een van weinige keren dat een DBA blij zal zijn met een ORA-00600. De oplossing gaat nu als volgt:

```
SQL> shutdown abort <bt>
```

(eigenlijk overbodig want de instance is al gecrasht)

```
SQL> startup mount
SQL> alter session set events '10015 trace name adjust_scn level 1';
SQL> alter database open;
```

De database is nu weer open! De redologfiles zijn er inderdaad weer alle drie, en van de goede grootte (102401 KB) en met de juiste status, dus geen 'INVALIDATED' meer. Verder blijken de objecten in de database benaderbaar te zijn, ook dat is een hele geruststelling.

Nu eerst de data veiligstellen, voor straks in de nieuwe database. De export loopt echter fout met een

```
ORA-24324: service handle not initialised
```

Het gaat dus nog niet helemaal goed met de database.

Nu de tweede grote test voor deze database. Gaat hij ook open zonder de aanpassing in de `initEEN.ora`? Met andere woorden, de hidden parameters worden verwijderd en we zetten Automatic Undo weer aan. De database start keurig. De export gaat helaas mis. Er verschijnt een

```
ORA-01092: ORACLE instance terminated. Disconnection forced.
```

Een crash dus. In de trace file komt ook een ORA-00600 [4194] voor, duidend op corrupte UNDO.

Lukt een herstart zonder de hidden `_ALLOW_REDOLOGS_CORRUPTION` maar mét Manual Undo wel? Ja, de database opent en alles is benaderbaar. En de export lukt nu gelukkig wel. Oracle

Customer Support meldt het volgende op de vraag of de database met Automatic Undo weer netjes open en stabiel te krijgen is: *“Once these parameters have been used to open a database, then the database is unsupported, regardless of whether it is then reopened without the parameters in.*

Therefore, the only option is to export and rebuild.”

Verder werd gemeld dat dit vermelde gedrag *expected behaviour* is.

Met de export en de bewaard gebleven database creatie-scripts maken we een nieuwe database. De database wordt gestopt, gevolgd door het opnieuw maken van een koude back-up en het verwijderen van de oude EEN-database. Met oradim verwijderen we ook de windows-service en maken de nieuwe database aan. De redologfiles worden nu wel gemirrored, op de D:- én de F:-schijf. Ook verspreiden we de controlfiles over twee schijven.

Als laatste stap voor EEN volgt import van de data. Dit gaat geheel naar wens. Ergo: database EEN is hersteld.

TWEE

Database TWEE blijkt makkelijker te herstellen. Deze database is robuuster aangemaakt, met gemirrored redologfiles. Er is weliswaar een complete mirror verdwenen van de D:-schijf, maar de set op de F:-schijf is nog intact. De database draait dus gewoon door.

Even twee nieuwe redologgroepen (single logfiles) aangemaakt als tussenoplossing, daarna de originele groepen verwijderd en opnieuw gemirrored aangemaakt. Daarna de tijdelijke extra redologgroepen weer verwijderd, en alles is weer in orde.

DRIE

Database DRIE ziet er net zo uit als EEN, dus hierbij volgen we het complete herstelscenario zoals bij EEN. Weer starten met de hidden parameters in de initDRIE.ora, doen alsof we recoveren en proberen de database te openen. Maar wederom verschijnen een ORA-03113 en een ORA-00600 [2662]. Het commando

```
SQL> alter session set events '10015 trace name adjust_scn level 1';
```

werkt niet. De database wil niet openen, sterker nog, hij crasht. Volgens de note-instructie mogen we het level verhogen (tot maximaal 6, komt de database dan nog niet op, dan is de database onherstelbaar beschadigd). Na de startup mount volgt dan het volgende commando:

```
SQL> alter session set events '10015 trace name adjust_scn level 2';
```

Nu opent de database wel, alle objecten zijn benaderbaar. Weer herstarten zonder `_ALLOW_REDOLOGS_CORRUPTION` maar met

Manual Undo, en er wordt succesvol een Entire Database export gemaakt.

Omdat de D:-schijf redelijk vol zit (dat was immers de aanleiding tot de schoning) en de F:-schijf over nog heel veel vrije ruimte beschikt, nemen we de beslissing om te proberen de nieuw aan te maken database op de F:-schijf te zetten. Niet alle applicatie-tablespace creatiescripts zijn aanwezig, en de export-file is te groot om te bewerken en zo de creatiescripts er uit te halen. We kiezen dus voor een tussenoplossing: van alle tablespaces waar de creatiescripts aanwezig zijn, zullen de datafiles op de F:-schijf komen, de rest eventueel achteraf.

De oude DRIE wordt weer veiliggesteld, en een nieuwe DRIE aangemaakt. De import gaat goed, en ook database DRIE is hersteld. Uiteraard weer met gemirrored en gespreide redologfiles en gespreide controlfiles.

Het blijkt dat de D:-schijf weer voldoende vrije ruimte heeft, waarop we besluiten de datafiles niet te gaan verplaatsen.

VIER

Database VIER lijkt qua herstel weer sterk op DRIE. Er is een `ADJUST_SCN LEVEL 2` nodig om de database open te krijgen. Alles gaat dan ook goed totdat de Entire Database export gemaakt gaat worden. Deze faalt keer op keer met de volgende foutmelding (en een core dump):

```
ORA-03113: End-of-file on communication channel. Disconnection forced.
ORA-01041: internal error. hostdef extension doesn't exist.
```

Oracle Customer Support geeft aan dat wellicht de `LARGE_POOL` te klein is. Het lijkt niet de oorzaak, want hij is niet kleiner dan bij de andere databases, maar het is tenminste het proberen waard. Het helpt echter niet. Het tweede alternatief van Oracle Customer Support, na het checken en goedbevinden van een aantal controles, is het maken van user-exports.

De controles zijn de volgende:

- Is DUAL benaderbaar, levert een select één rij op, bestaat er één DUAL?
- Zijn alle objecten van SYS valid?
- Zijn alle tablespaces online?

Alle vragen worden met JA beantwoord, dus proberen we de user-exports. Als dat lukt betekent het wel veel extra werk, maar dan is tenminste alle data terug. Dit leidt dan ook tot wat meer voorbereidend werk. Van te voren moeten immers alle creatiescripts klaar staan voor de rollen, users (inclusief alle rechten en privileges) en tablespaces. Ook voor deze database blijkt dat er geen applicatie-tablespace creatiescripts bewaard zijn.

Er is geen TOAD, SQL*Developer of iets dergelijks ter beschikking, maar vanaf Oracle 9i bestaat er natuurlijk wel een heel

handig middel in de vorm van het DBMS_METADATA package. Hiermee worden alle scripts met behulp van de volgende commando's gegenereerd:

```
SQL> select dbms_metadata.get_ddl('USER', '<usernaam>') from dual;
SQL> select dbms_metadata.get_ddl('ROLE', '<rolnaam>') from dual;
SQL> select dbms_metadata.get_ddl('TABLESPACE', '<tablespacenaam>') from dual;
SQL> select dbms_metadata.get_granted_ddl('USER', '<usernaam>') from dual;
```

Bijkomend voordeel is dat de originele wachtwoorden bewaard blijven.

Op www.oraclealchemist.com/news/trick_1_copying_users_the_right_way is een handig script te vinden om dit voor users in één keer te genereren. Je hoeft de output van dit script alleen maar te voorzien van “/” of “;” op de juiste plaatsen. Het maken van de user-exports voor alle niet-Oracle-eigen users gaat gelukkig wel goed, dus maken we een begin met het definitieve herstel. Voor alle zekerheid wordt een koude backup van de oude VIER gemaakt om daarna de oude VIER te verwijderen.

De nieuwe VIER wordt kaal aangemaakt, de volgende stap is het aanmaken van de tablespaces, rollen en users. Van de negen user-imports verlopen er zes feilloos. Twee gaan fout op ontbrekende system-privileges en ontbrekende quota op tablespaces. De user-creatiescripts blijken dus niet helemaal volledig te zijn. Deze privileges en quota zijn makkelijk alsnog uit te delen, en de imports gaan alsnog goed. Er verschijnen wel warnings over invalid objects (compilation warnings), maar dat is makkelijk uit te zoeken en tevens van later zorg. De negende import baart immers wat meer zorgen, en is volgens de klant juist de belangrijkste user. Voor een zevental tabellen komt de volgende melding:

```
IMP-00060 Warning: Skipping table OWF_MGR."tabelnaam" because object type SYSTEM."naam" does not exist or has different identifier.
```

Het blijkt dat applicatie-eigen objecttypes, aangemaakt in het schema van SYSTEM, ontbreken. Die zullen ongetwijfeld in de oude VIER terug te vinden zijn, maar het kost weer kostbare tijd om die helemaal weer terug te zetten. Gelukkig komt de klant met de mededeling dat er nog een oude lege kopie van de VIER op een andere databaseserver staat. Als de structuur van de applicatie compleet is, betekent dat de redding. Het blijkt dat die oude kopie helemaal compleet is, weliswaar zonder applicatie-data, maar die staan toch al in de exports. Aan de hand van die kopie kan dus de applicatie-eigen objecttypes van SYSTEM worden gereconstrueerd. Vervolgens kunnen we de ontbrekende tabellen weer importeren. Tot grote verbazing treedt weer de IMP-00060 op. Nazoeken op Metalink leert dat dit komt door een bug in de imp-utility, de workaround

is eenvoudig. Maak de tabellen eerst handmatig aan, en importeer dan de data. Uit de dmp-file door middel van de optie `imp indexfile=<naam>` de objectcreatie statements gehaald en dit bewerkt tot een uitvoerbaar script. Tevens blijkt dat de bewuste tabellen geen rijen bevatten, dus een import achteraf van die data kan achterwege blijven. Het aanmaken van de tabellen, inclusief indexen en constraints, verloopt zonder fouten. De voor de tweede keer optredende IMP-00060, terwijl de objecttypes er wel degelijk zijn, ligt dus inderdaad aan de bug.

Nu nog alle objecten VALID zien te krijgen. Van drie users zijn honderden objecten INVALID. Het draaien van utlrp brengt het aantal invalid objecten terug tot zestig. Van 58 objecten blijkt dat select-rechten op een aantal v\$_-views te zijn. Select-rechten op de v_\$_-objecten uitgedeeld, en die 58 objecten zijn ook weer valid.

Een van weinige keren dat een DBA blij zal zijn met een ORA-00600

Er blijven dus twee invalid objecten over, twee package bodies. Eén package body geeft een syntax-error bij het compileren. In de oude kopie-database is de package body valid, dus het creatiestatement daaruit gevist en op VIER gedraaid. Het resultaat is een valid object. De laatste invalid package body meldt een niet-bekende bron (WF_ALL_JOBS) bij het compileren. Op zoek naar WF_ALL_JOBS op VIER. Niet te vinden – niet als tabel, niet als view, niet als synoniem, etc. In de kopie-database gezocht, en hij is te vinden als een applicatie-eigen view (+public synonym) in het schema van SYS. De definitie naar VIER overgehaald, en de laatste package body is ook valid.

Ten slotte

Het heeft bijna een volle werkweek geduurd om de databases na een simpele foute actie weer te herstellen. Het is dus echt belangrijk om een database robuust op te bouwen. Oracle adviseert niet voor niets om redologfiles te mirrorren en te spreiden.

Ook het belang van een goede back-up is weer eens onderstreept. Altijd handig en nuttig is het bewaren en up-to-date houden van alle creatiescripts. Zo kunnen we de database, alle tablespaces en rollen/users en eventueel de gehele applicatie makkelijk en snel aanmaken als je van scratch moet opbouwen. Opmerkelijk is dat schijnbaar hetzelfde probleem leidt tot (in detail) drie verschillende wijzen van herstel.

Eddy Jongejans en **Robin Boëtius** (met resp. 23 jaar en 4 jaar Oracle-ervaring) zijn als DBA werkzaam bij Transfer Solutions.