Database horror

Het reanimeren van een 'fool back-up'

Soms is het leven niet mooi. Laatst werd mij gevraagd om een database te reanimeren waarvan alleen een back-up bestaat zonder dat deze in back-up modus staat. Ja, vandaar de 'fool back-up' (het tegenovergestelde van een 'full back-up', een online back-up met de database in back-up mode). Het betreft Windows 2003, database Oracle EE 9.2.0.7.

Is dit mogelijk? Ja.
Zou je dit willen? In vrijwel ieder geval: Nee!
Raak je data kwijt? Ja, afhankelijk van de activiteit op dat moment.

Actie

Voor de geïnteresseerde lezer: de Oracle-database heeft een ongedocumenteerde parameter welke specifiek gemaakt is om databases met datafiles welke niet allemaal consistent zijn, of databases met datafiles welke 'fuzzy' zijn (de 'fuzzy bit' is een vlag in de fileheader van een oracle datafile, welke oracle gebruikt om de status van de file in neer te zetten. Omdat de 'fuzzy bit' gezet wordt tijdens het in back-up modus zetten voor een back-up, weet de database dus dat deze in back-up modus was, en gerenoveerd kan worden) op te kunnen brengen. Deze parameter is '_allow_resetlogs_corruption'. Het advies van Oracle-support (en dat van mij) is: gebruik deze parameter alleen wanneer Oracle-support dit voorstelt. In eerste instantie was het probleem dat een paar datafiles vernietigd waren. Dit leidde tot de eerste actie: alleen de vernietigde datafiles restoren (deze datafiles waren echter verkeerd geback-upt), en proberen de database open te forceren. (de database was echter drie dagen ouder dan de gerestorede datafiles).

Datafiles

Deze actie leidde tot een ORA-1555 (snapshot too old) bij het openen van de database (waarbij ook de instance verdwenen was). Nader onderzoek bracht aan het licht dat dit kwam door de locally managed datafiles (de tracefile liet een query zien van de database op UET$ (de 'used extents' data dictionary view), welke eerst een (geclusterde) tabel was in de SYSTEM tablespace, maar met locally managed datafiles feitelijk zijn data uit de datafile bitmap area haalt, waarin bij locally managed tablespaces de ruimte-allocatie wordt bijgehouden). Dit betekent dus dat een bepaalde 'versie' van een bitmap block in de datafile niet gevonden kan worden, wat niet zo verwonderlijk is wanneer er datafiles zijn die drie dagen verschillen van de rest van de database.

Inconsistent

Volgende actie: een totale restore van de 'fool back-up'. Het openen van de database werkte hierbij (met de boven genoemde parameter gezet, uiteraard)! Interessant! De volgende actie is (uiteraard) is om alle user-data uit de database te exporteren (om de data te redden, de fysieke database zelf is 100 procent zeker inconsistent en onbruikbaar!)

Het resultaat dat we bereikten, had ook op verschillende andere manieren verwezenlijkt kunnen worden

We zijn ons er van bewust dat we inconsistente data kunnen hebben, en ook corrupte blokken tegen kunnen komen. Om ook dit probleem op te kunnen lossen, hebben we een andere ongedocumenteerde parameter gezet in de parameterfile, zodat de database niet crasht, maar de corrupte blokken overslaat als hij deze tegenkomt. (van tabellen, een export leest geen index data tijdens de export, maar haalt de creatie informatie voor de indexen uit de data dictionary) Dit event is 10231, en moet gezet worden in de parameter file:

```
event = '10231 trace name context forever, level
10'
```

Hiermee hebben de meeste tabellen kunnen exporteren, slechts enkele gaven een ORA-1555 (wat geen corruptie is, dus het event helpt hierbij niet). De ORA-1555 betekent dat de 'before image' van een block niet aanwezig is op het 'UBA' (undo block address).
Elk index block en elk data block heeft een lijst met transacties in de variable transactie header van het block. (default, tijdens creatie is de lijst 1 groot voor data en 2 voor indexblokken). De naam van deze lijst is 'ITL', wat 'interested transaction list' betekend. Als een transactie uitgevoerd wordt, zorgt deze lijst voor consistente reads voor andere transacties. Als een leesactie de UBA in de ITL moet gebruiken (omdat de leesactie consistent is met de 'before image'), zoekt dit process op de het address in de UBA (wat een plaats in het undo segment is). Als het process toch zijn data daar niet kan vinden, komt de melding ORA-1555.

Vlaggen

Met deze melding hebben we een fantastische 'logical corruption': de database snapt de data in het blok helemaal (dus het is niet fysiek corrupt), maar kan de actie niet uitvoeren, omdat het niet de consistente data kan vinden voor de gevraagde actie. Op dit punt hebben we besloten er uit te halen wat we konden, door de blokken corrupt te vlaggen (zodat het event 10231 er voor zorgde dat deze overgeslagen worden). Grof?

We zijn ons er van bewust dat we inconsistente data kunnen hebben, en ook corrupte blokken tegen kunnen komen

Dat klopt, maar het zorgde er voor dat we de meeste data er uit konden halen…
Dit hebben we gedaan door de BBED ('block browser and editor') te gebruiken. Het is een commandline tool, welke default aanwezig is op Windows, en gelinkt kan worden tot een executable op Linux/Unix. Het is ook beveiligd met een wachtwoord. De BBED kan op een blok 'gezet' worden, en deze met het commando 'corrupt' corrupt vlaggen (wat betekent dat een vlag in het blok gezet wordt om aan te geven dat het medium corrupt is). Het reeds gezette event 10231 zorgt er dan voor dat het blok overgeslagen wordt.
Om achter de DBA's (data block addresses) te komen van de 'probleemblokken', hebben we het volgende event gezet (zodat het duidelijk is welk blok we moeten hebben):

```
event = '1555 tracee name errorstack level 10'
```

Dit heeft uiteindelijk er toe geleid dat we bijna alle data uit deze database konden halen.

Hergenereren

Vanwege de manier waarop de uiteindelijke rijen uit een blok te lezen zijn met de BBED, is het ook duidelijk geworden dat dat met de BBED niet goed mogelijk is (de tool is er ook niet voor gemaakt, het is gemaakt om een blok te kunnen onderzoeken, en eventueel aanpassingen te kunnen doen), hiervoor zijn echter tools zoals Oracle's DUL, en ORA600's DUDE/jDUL. Het resultaat dat we bereikten, had ook op verschillende andere manieren verwezenlijkt kunnen worden. De keuzen zijn steeds strikt gemaakt om data met de voor de ontwikkelaars beschikbare hulpmiddelen te kunnen hergenereren, en om daarbij het risico zo klein mogelijk te houden (met een inconsistente database).

Frits Hoogland is consultant bij LogicaCMG, en houdt zich bezig met Oracle- en andere databases, applicatieservers en webtechnologie.