Database fragmentatie

Verbeterde mogelijkheden in Oracle 9i

Fragmentatie van een database is één van de meest voorkomende problemen die een database administrator kan tegenkomen. Het vinden en vervolgens oplossen van fragmentatie is een tijdrovende klus. De resources die gebruikt worden om fragmentatie zoveel mogelijk te voorkomen, zullen zich in een veelvoud terugbetalen in de verminderde tijd die wordt besteed aan het beheer van de database. Dit artikel gaat in op de mogelijkheden die door Oracle worden geboden om fragmentatie op te lossen en vervolgens te voorkomen en is gebaseerd op de versies 8.1.x en 9i.

Een database bestaat primair uit een collectie van bestanden die "datafiles" worden genoemd. In deze datafiles bevinden zich de database-objecten zoals tabellen, indexen en clusters. Om al deze objecten te ordenen heeft Oracle de volgende methodiek bedacht: op het laagste niveau bevindt zich het datablok. Deze zijn vervolgens gegroepeerd in extents. Extents zijn een set van aaneen gesloten databloks. Extents worden door Oracle logisch gezien per type in een segment gegroepeerd. Een segment is dus een verzameling extents van bijvoorbeeld het type tabel, index of rollback. Een tablespace op zijn beurt bestaat uit één of meerdere datafiles.

Fragmentatie van een tablespace ontstaat doordat een object (tabel of index) meerdere extents gebruikt die niet aaneengesloten in de datafile zijn opgeslagen. Gemiddeld zal een gefragmenteerde tablespace een performance-degradatie van 10 tot 20 procent van de i/o call tot gevolg hebben. Chaining van rijen kan zelfs een degradatie tot 50 procent van de i/o call tot gevolg hebben.

Het benaderen van meerdere datablokken kan een behoorlijke i/o performance dip geven

Op het gebied van extent management zijn er vanaf Oracle RDBMS versie 8i twee soorten tablespaces:

- *Locally managed tablespaces:* de tablespace houdt zelf middels een bitmap de vrije en gebruikte extents bij. De wijzigingen worden door Oracle rechtstreeks in de bitmaps bijgehouden. Er wordt dus geen rollback gegenereerd.
- *Dictionary managed tablespaces:* dit is tot en met 8.1.x de default setting van space management. Ruimtegebruik van een tablespace wordt in de data-dictionary bijgehouden. Dit is de legacy methode van space management in de Oracle database.

Om fragmentatie te voorkomen of op te lossen kunnen de volgende stappen uitgevoerd worden:

- gebruik de juiste tabelgrootte;
- creëer een nieuwe tablespace en verplaats de data hier naar toe;
- maak een export met compress = y, drop de tabel en importeer deze;
- verhoog de grootte van het volgende extent;
- voorkom chaining door de percentages van PCTFREE en PCTUSED goed te zetten;
- in het ergste geval zal de gehele database opnieuw moeten worden opgebouwd , met eventueel een andere data- blok -size en vervolgens - in zijn geheel moeten worden geïmporteerd.

Voor het vinden van fragmentatie kan men een van de volgende scripts draaien:

```
select   tablespace_name,sum(bytes)/1024/1024 total_free
from     dba_free_space
group by tablespace_name;
```

of

```
select segment_name, segment_type, extents, bytes
from dba_segments
where extents > 5;
```

SEGMENT_NAME	SEGMENT_TYPE	EXTENTS	BYTES
WOZ_I_ST30_1	INDEX	94	1605632
WOZ_I_ST30_2	INDEX	14	294912
WOZ_P_ST30	INDEX	115	966656
WOZ_I_ST40	INDEX	202	3375104
WOZ_I_ST40_1	INDEX	267	4440064
WOZ_I_ST40_2	INDEX	179	3072000
WOZ_I_ST40_3	INDEX	374	6266880
WOZ_P_ST40_4	INDEX	305	2523136

Om een tablespace grafisch inzichtelijk te maken is er de tablespace map utility in de Enterprise Manager console. Zie figuur 1 voor een screendump van deze tool.

Fragmentatie ontstaat doordat een tabel groeit en vervolgens meerdere extents gaat vullen. Dit komt omdat er rijen worden toegevoegd of omdat er rijen worden ge-update met data die niet in de oorspronkelijke rij in het datablok passen (chaining). Om fragmentatie te voorkomen dient men te proberen de grootte van een object goed in te schatten. Als een tabel al is gevuld is dit eenvoudig te doen. Als een segment van een tabel tien extents van 1mb heeft, dient een nieuwe tabel met 10 * 1 mb = 10mb gemaakt te worden. Dit doet men als volgt:

```
CREATE TABLE CUSTOMER1
TABLESPACE NEW
STORAGE (INITIAL 10M NEXT 5M PCTINCREASE 0)
AS SELECT * FROM CUSTOMER;
```

Vervolgens:

```
RENAME CUSTOMER1 TO CUSTOMER;
```

Tot slot dienen de indexen weer opnieuw opgebouwd te worden. Om rollback problemen te voorkomen, kan men NOLOGGING (v8.x) of UNRECOVERABLE (v7.2.x) aan het create table statement toevoegen.

Indien de fragmentatie meerdere tabellen en indexen betreft is het soms beter een export te maken met de parameter COMPRESS = Y. Vervolgens dienen de tabellen verwijderd en opnieuw geïmporteerd te worden. Dit heeft als belangrijk nadeel ten opzichte van de vorige methode, dat de data voor een langere periode niet benaderbaar zijn. Als men geen ruimte of tijd heeft om fragmentatie meteen op te lossen kan men voor de probleemgevallen de next extent parameter van een object aanpassen om nog meer fragmentatie te voorkomen. Dit gaat als volgt: ALTER TABLE CUSTOMER STORAGE (NEXT 35M);

Locally Managed Tablespaces

Vanaf Oracle 8i kunnen tablespaces dictionary of locally managed worden. Dit geldt voor alle tablespaces behalve die van SYSTEM. In een locally managed tablespace wordt het ruimte gebruik in een bitmap opgeslagen. Dit is bij default in de eerste 64KB van de tablespace. Eventuele extra bitmap segmenten worden hieraan toegevoegd. Vanaf Oracle 9i worden de tablespaces default locally managed aangemaakt. In iedere datafile van de tablespace wordt er een bitmap bijgehouden. Hierin staan de vrije en gebruikte data-bloks van de datafile. Iedere keer dat er een extent vrij komt dan wel gealloceerd wordt de bitmap ge-update. Doordat er geen wijzigingen in de data dictionary worden uitgevoerd is er geen rollback activiteit en dit is weer voordelig voor onder andere de performance van de database. Tevens is er geen reden meer om de vrije extents samen te voegen (coalesce) omdat de vrije ruimte direct in de bitmap wordt bijgehouden. In een locally managed tablespace kunnen alle extents dezelfde grootte hebben.

Als je een tablespace locally managed maakt is het niet mogelijk om de waarden voor de default storage clause, NEXT, PCTINCREASE, MINEXTENTS en MAXEXTENTS mee te geven. Locally managed tablespaces hebben minder last van fragmentatie en de objecten in zo'n tablespace ervaren minder ruimte gerelateerde problemen.

Om een locally managed tablespace aan te maken is er de volgende toevoeging aan het CREATE TABLESPACE commando toegevoegd:

```
CREATE TABLESPACE tbs1 DATAFILE 'tbs1_01.dbf'
         EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
```

De EXTENT_MANAGEMENT_CLAUSE heeft de volgende mogelijkheden:

```
DICTIONARY | LOCAL
        AUTOALLOCATE | UNIFORM [SIZE integer [K|M]]
```

DICTIONARY	De tablespace wordt gemanaged in de datadictionary
LOCAL	De tablespace wordt lokaal gemanaged met een bitmap
AUTOALLOCATE	De tablespace wordt op systeem niveau gemanaged. Gebruikers kunnen niet de extent grootte bepalen

UNIFORM — De tablespace wordt gemanaged met gelijke extents van SIZE bytes grootte in M megabyte of K kilobytes. Default is de extent size 1 megabyte

Chaining

Chaining wordt veroorzaakt indien er niet genoeg ruimte is in een datablok om wijzigingen op te slaan. Een chained record is een rij die in meerdere datablokken voor komt. Het benaderen van meerdere datablokken kan een behoorlijke i/o performance dip geven. Er is een aantal methoden om te zien of er chaining in de database bestaat. Ten eerste is er een script $ORACLE_HOME/rdbms/admin/utlchain.sql. Dit script vult de chained_rows tabel;

```
@$ORACLE_HOME/rdbms/admin/utlchain.sql
select HEAD_ROWID
from CHAINED_ROWS
where TABLE_NAME = 'CUSTOMER';
```

De tweede methode is het analyseren van een tabel met de volgende syntax;

```
ANALYZE TABLE CUSTOMER LIST CHAINED ROWS;
```

In de view dba_tables kan dan worden gekeken over er tabellen zijn met chaining:

```
select TABLE_NAME, OWNER, CHAIN_CNT, NUM_ROWS
from DBA_TABLES
where CHAIN_CNT > 0;
```

Als er geen rijen worden teruggegeven is er geen chaining probleem. Als er wel een probleem is, dient men de PCTFREE parameter van de tabel te verhogen, default staat deze op tien procent. Voor tabellen met een hoog aantal wijzigingen dient de waarde te worden verhoogd. Er zal dan geen verdere chaining meer ontstaan. Om de huidige aantal chained records te verminderen is een export, drop van het object en vervolgens een import noodzakelijk.

Data dictionary fragmentatie

Iedere keer dat er een object wordt aangemaakt of gewijzigd in de database worden er verschillende interne tabellen aangepast. Dit heeft tot gevolg dat de data dictionary gefragmenteerd kan raken. De kans hierop wordt groter naarmate het aantal objecten toeneemt. Helaas is er maar een mogelijkheid om deze vorm van fragmentatie op te lossen: het opnieuw opbouwen van een database. Let op: dit kan veel tijd in beslag nemen en zal daarom goed moeten worden gepland. Om een database opnieuw op te bouwen dienen de volgende stappen doorlopen te worden;

- Volledige database export;
- Complete back-up van de datafiles, control files, online redo log files en de init.ora;
- Creëer een nieuwe database met het create database script;
- Zorg voor een groot genoeg rollback segment voor de import;
- Importeer alle objecten en data.

Oracle Tuning Pack 2.0 Tablespace viewer

De tablespace viewer (onderdeel van Tablespace Manager) geeft een grafische voorstelling van alle tablespaces, data- files, segmenten, totaal aantal datablokken, vrije data- blokken en het percentage vrije data blokken in een tablespace. Indien noodzakelijk is er een reorganisatie wizard die automatisch problematische objecten kan repareren. Deze wizard maakt gebruik van de export- en import-tools van Oracle. Het is ook mogelijk om met deze wizard segmenten te verplaatsen naar andere tablespaces.

Automatic Segment Space Management (v9.0.1)

In versies van het Oracle RDBMS voor 9i waren er freelists voor het bijhouden van data- blokken binnen een database object dat ruimte heeft voor een nieuwe rij. Hier wordt bij het creëren van een object aangegeven middels de PCTUSED wanneer een datablok op die freelist komt. PCTUSED bepaalt zo de 'vullingsgraad' van een datablok. Als een datablok minder gevuld is dan het percentage van de PCTUSED parameter, staat een datablok op de freelist.

In het geval van automatic segment space management worden er bitmaps gebruikt in plaats van deze freelists. Bitmaps geven aan hoeveel vrije ruimte er nog is voor dit object in deze datablok. Hierbij wordt onderscheid gemaakt tussen meer dan 75 procent, tussen de 50 en 75 procent, tussen de 25 en 50 procent of min-

Figuur 1. Screendump van de tablespace map utility

der dan 25 procent. Deze nieuwe toepassing zorgt ervoor dat er geen tuning meer nodig is voor de DBA op het gebied van ruimtegebruik van de objecten in de tablespace. Tevens zal dit ervoor zorgen dat er minder last van fragmentatie is. Er wordt door het gebruik van de bitmaps een betere 'vullingsgraad' bereikt en de performance bij DML zal verbeteren omdat er tegelijkertijd verschillende delen van de bitmap geraadpleegd kunnen worden voor de vrije ruimte.

Automatic space management kan alleen worden gebruikt in locally managed tablespaces. Alle objecten in de tablespace zullen gebruik maken van automatic space management. Bij objecten die worden gecreëerd in zo'n specifieke tablespace worden de specificaties voor PCTUSED, FREELISTS en FREELISTGROUPS genegeerd. In de view DBA_TABLESPACES is _ een nieuwe column SEGMENT_SPACE_MANAGEMENT toegevoegd die aangeeft in welke mode een tablespace staat. De initialisatie parameter compatible moet op minimaal 9.0.0 worden gezet.

Conclusie

De mogelijkheden van de RDBMS om fragmentatie tegen te gaan, zijn in de nieuwste releases van de Oracle sterk verbeterd. Toch zal er nog altijd door een DBA de nodige tijd moeten worden genomen om te zorgen voor een optimaal space management. Ook is het belangrijk om bij het ontwerp en de bouw van een database met fragmentatie rekening te houden. Er kan bijvoorbeeld gebruik worden gemaakt van tablespaces die locally managed zijn met verschillende uniform extent size. Er kan dan onderscheid worden gemaakt tussen kleine, middel en grote database objecten. Voor VLDB is het zeker aan te bevelen om de tablespaces locally managed te maken.

Literatuur

Titel: Oracle Performance Tuning
Auteur: Richard Niemiec
Uitgeverij: Oracle Press
ISBN: 0-07-882434-6
Note 105120.1 Advantages of using locally managed vs dictionary managed tablespaces
Note 93771.1 Locally managed tablespaces in Oracle 8i
Paper #711 How to stop defragmenting and start living: The definitive word on fragmentation

Niels Zegveld
is werkzaam als DBA-consultant bij Caesar DBA in Utrecht. Hij adviseert organisaties in het gebruik van Oracle RDBMS producten. Caesar DBA is een onderdeel van Caesar Groep te Utrecht.

44: Array-stopper