

Datapump Specials

Marco Patzwahl
MuniQSoft GmbH
München-Unterhaching

Schlüsselworte:

Datapump, Tuning, Tipps & Tricks

Einleitung

Jeder Oracle Administrator musste schon mal einen Export oder Import seiner Oracle Daten durchführen und hat dabei vermutlich schon das ein oder andere Problem erlebt. Dieser Vortrag soll Ihnen einige Tipps & Tricks aufzeigen, die den Arbeitsalltag (hoffentlich) verbessern können.

Blockkorruption

Ein Datapump Export findet keine Blockkorruption und überspringt kommentarlos defekte Blöcke. Das bedeutet, dass unter Umständen Zeilen kommentarlos beim Export verschwinden und auch nicht wieder gerettet werden können.

Der "alte" Export bemerkt eine Blockkorruption und stürzt ab.

Workaround für Benutzer des expdp:

Regelmäßig mit dem Oracle RMAN Tool ein Backup durchführen.

Den alten Export verwenden und Datei auf /dev/null legen. Hier ist dann nur das Logfile interessant.

Den Prüf-Befehl verwenden: ANALYZE TABLE/INDEX VALIDATE STRUCTURE

Das Oracle DB Verify Utility (dbv) einsetzen, denn auch das erkennt eine Blockkorruption.

Wer schon die Version 11.1 oder 11.2 einsetzt, kann auch für bestimmte Objekte einen Health-Check durchführen.

Konsistenz

Damit der Export konsistent ist, muss der Parameter FLASHBACK_TIME (bzw. FLASHBACK_SCN) verwendet werden.

In Version 10.1-11.1 verwendet man den Parameter FLASHBACK_TIME:

```
expdp userid=system/sys ...  
flashback_time=\"to_timestamp(  
to_char(systimestamp, 'DD/MM/YYYY HH24:MI:SS.FF'),  
'DD/MM/YYYY HH24:MI:SS.FF')\"
```

Damit wird ein konsistenter Export zur aktuellen Uhrzeit durchgeführt. Sie können natürlich auch eine andere Zeit (z.B. vor 5 Minuten oder auch 23:30:32) verwenden.

Ab Version 11.2 kann via Legacy Mode (alte Parameter des exp werden einfach in ihr aktuelles Pendant umgewandelt) auch der Parameter CONSISTENT benutzt werden:

```
expdp userid=system/sys ...  
consistent=y
```

Benutzer/Rechte

Benutzer SYS:

Benutzer mit SYSDBA Rechten sollten nicht für expdp/impdp verwendet werden.

Wer jedoch (in Absprache mit dem Support) mit dem Benutzer SYS exportieren oder importieren möchte, verwendet einfache Hochkommata für den Benutzer und damit folgende Syntax:

```
expdp userid='/ as sysdba' directory=data_pump_dir schemas=scott
dumpfile=scott.dmp
```

Benutzer SYSTEM und weitere Benutzer...

dürfen exportieren, wenn sie die folgenden Rechte/Rollen besitzen:

CREATE SESSION

CREATE TABLE (für die Datapump Master Tabelle)

EXP_FULL_DATABASE (für alle Objekte)

WRITE Recht auf dem Directory, in das exportiert werden soll. DBA Rolle oder FLASHBACK Recht auf allen XML Objekten, die einem nicht gehören.

Sie vergeben die notwendigen Rechte z.B. durch:

```
GRANT read, write ON directory data_pump_dir TO scott
```

```
GRANT create session TO scott;
```

```
GRANT create table TO scott;
```

```
GRANT flashback on xdb.my_xml_tab TO scott;
```

Für den Import sind folgende Rechte notwendig:

CREATE SESSION

CREATE TABLE

entsprechende Rechte für alle zu erzeugenden Objekte oder DBA Rolle oder alternativ die IMP_FULL_DATABASE Rolle (für alle Objekte)

READ Recht auf dem Directory, aus dem gelesen werden soll. Beachten Sie, dass auch bei einem Import ein Write Recht auf einem Directory bestehen muss, in das das Logfile gelegt wird.

Parameter: SAMPLE

SAMPLE wählt nur einen prozentualen Querschnitt aus der Tabelle für den Export aus.

Das funktioniert nur, wenn keine PK/FK Beziehungen zwischen den Tabellen bestehen, weil Oracle willkürlich Daten herausnimmt.

Die Daten passen sonst beim Import nicht mehr zusammen.

Beispiel (nur 1% der Daten exportieren):

```
expdp userid=system/sys directory=data_pump_dir dumpfile=marco.dbf SAMPLE=1
```

Parameter: QUERY

Mit dem Parameter Query können nur bestimmte Zeilen der Tabellen exportiert werden.

Beispiele:

Query Klausel für mehrere Tabellen:

```
QUERY= emp:\ "WHERE deptno= 10\ "
```

```
QUERY=dept:\ "WHERE deptno= 10\ "
```

Wie kann man seine Tabelle beim Export sortieren?

```
QUERY= scott.emp:\ "ORDER BY sal\ "
```

Nur Blöcke mit geänderten Zeilen exportieren:

```
QUERY="scott.emp:\ "WHERE ora_rowscn>12345678\ "
```

Beispiel (bei Windows bedeutet ^: es geht in der nächsten Zeile weiter):

```
expdp userid=scott/tiger directory=data_pump_dir dumpfile=scott.dmp ^
reuse_dumpfiles=y ^
query=emp:\ "where deptno=10\ " ^
query=dept:\ "where deptno=10\ "
Ausgabe (gekürzt)
. . "SCOTT"."DEPT"          5.875 KB          1 Zeilen exportiert
. . "SCOTT"."EMP"          8.125 KB          3 Zeilen exportiert
```

Parameter: NETWORK_LINK

Ein Network Link überträgt die Daten von einer Datenbank mittels DB Link zu einer anderen Datenbank, ohne physische Dump-Dateien im Betriebssystem anzulegen.

Voraussetzungen:

Datenbank Link für Export/Import Benutzer vorhanden.

Wenn Log-Files erzeugt werden sollen, ein Directory.

Hinweis:

Tabellen mit Long Spalten können nicht übertragen werden.

Vorbereitung:

Database Link erstellen

```
CREATE [PUBLIC] DATABASE LINK prod
CONNECT TO scott IDENTIFIED BY tiger
USING '172.20.30.107:1521/o11g';
```

Test des DB Links

```
SELECT * FROM dual@sl07;
```

Beispiele:

Tabellen übertragen:

```
impdp userid=scott/TIGER tables=(emp,dept) network_link=sl07 nologfile=y
table_exists_action=replace
```

Schema übertragen:

```
impdp userid=scott/TIGER schemas=scott network_link=sl07
table_exists_action=replace nologfile=y
```

DB komplett übertragen:

```
impdp userid=system/sys full=y network_link=sl07
table_exists_action=replace nologfile=y
```

Parameter: EXCLUDE

Mit dem Parameter können bestimmte Objekttypen oder Objekte vom Export ausgeschlossen werden.

Exclude mit IN Klausel:

```
EXCLUDE=TABLE:\ " IN ('EMP', 'DEPT')\ "
```

Exclude mit LIKE:

```
EXCLUDE=PROCEDURE:\\"LIKE 'MY_PROC_%'\\"
```

Exclude mit BETWEEN:

```
EXCLUDE=TABLE:\\" BETWEEN 'AAAAA' AND 'EZZZZZ'\\"
```

Exclude mit >:

```
EXCLUDE=TABLE:\\"> 'E'\\"
```

Scheduler Objekte ausschließen:

```
EXCLUDE=PROCOBJ:"IN ('SCHEDULE_TEST')"
```

Exclude mit =:

```
EXCLUDE=INDEX:"= 'PK_EMP'"
```

Alle Sequenzen ausschließen:

```
EXCLUDE=SEQUENCE
```

Hinweise:

EXCLUDE und INCLUDE dürfen nicht beim gleichen Datapump-Vorgang gleichzeitig verwendet werden. Eine Übersicht der zu filternden Objekte erhalten Sie über:

```
SELECT * FROM database_export_objects  
where named='Y';
```

Unter Windows müssen die "" entwertet werden mit \\"

Unter UNIX müssen auch die einfachen Quotes entwertet werden

Beispiel:

```
\\"EXCLUDE=TABLE:\\" =\'EMP\\'\\"
```

Alle Scheduler Objekte lassen sich derzeit nicht ausschließen, aber es gibt einen Enhancement Request: Bug 6472456 - Abstract: PRODUCT ENHANCEMENT: OBJECT_PATH FOR JOBS CREATED WITH DBMS_SCHEDULER.

Parameter: REMAP_TABLE

Mit dem Parameter REMAP_TABLE können Tabellen beim Import mit einem neuen Namen versehen werden.

Beispiel für Tabelle (Tabelle big in big_neu umbenennen):

```
impdp userid=x/y dumpfile=dp_dir:big.dmp REMAP_TABLE=scott.big:big_neu
```

Beispiel für Partition (Partition part in part_neu umbenennen):

```
impdp userid=x/y dumpfile=dp_dir:big.dmp  
REMAP_TABLE=scott.big.part:part_neu
```

Hinweis:

Tabellen können nicht umbenannt werden, wenn sie bereits existieren (auch wenn TABLE_EXISTS_ACTION auf TRUNCATE oder DROP steht)

Parameter: TRANSFORM

Der Parameter TRANSFORM kann die Segment Klausel beim Import manipulieren:

Beispiel:

Segment Attribute an-/abschalten

```
TRANSFORM=SEGMENT_ATTRIBUTES:[y|n]:TABLE
```

DEFERED SEGMENT CREATION an/aus:

```
TRANSFORM=SEGMENT_CREATION:[y|n]:TABLE
```

Datendateien und Extents auf 50 % der Größe setzen

```
TRANSFORM=PCTSPACE:50
```

Objekte aus verschiedenen Schemas:

Seit der Version 11.2 kann folgende Syntax verwendet werden:

```
TABLES=( scott.emp, marco.dept)
```

Bei Version 10.2 quitierte Oracle das noch mit:

UDE-00012: Bei Exportvorgängen im Tabellenmodus sind nur Objekte aus einem Schema zulässig

Workaround bis 11.1 (scott.emp,hr.employees,oe.categories.)

```
SCHEMAS=scott,hr,oe
```

```
INCLUDE=TABLE:\"IN('EMP','EMPLOYEES',  
'CATEGORIES_TAB')\"
```

Master Tabelle des Imports

Sie können eine Import-Simulation mit allen DDL Statements in eine Master-Tabelle schreiben lassen:

```
impdp userid=scott/tiger  
dumpfile=scott.dmp  
SQLFILE=sql.txt  
nologfile=y  
keep_master=y
```

...

Job "SCOTT"."SYS_SQL_FILE_FULL_03" erfolgreich um 16:15:03 abgeschlossen.

Ein Select auf die Tabelle offenbart die Schätze...

Hinweis:

Leider hat die Tabelle sehr viele Spalten.

Eine Auswahl könnte sein:

```
SELECT PROCESS_ORDER,  
DUPLICATE,  
DUMP_FILEID,  
DUMP_POSITION,  
DUMP_LENGTH,  
COMPLETED_ROWS,  
ERROR_COUNT,  
OBJECT_TYPE,  
IN_PROGRESS,  
OBJECT_NAME,  
OBJECT_SCHEMA,  
PARTITION_NAME,  
SUBPARTITION_NAME,
```

```
COMPLETION_TIME,  
OBJECT_TABLESPACE,  
SIZE_ESTIMATE,  
OBJECT_ROW,  
XML_CLOB,  
NAME,  
VALUE_T,  
start_time  
FROM SCOTT.SYS_SQL_FILE_FULL_03;
```

Alternativen zum Oracle Compress

Leider ist die Advanced Compression Option nur in der Enterprise Edition gegen Aufpreis verfügbar.

Oracle Compress:

Ersparnis: ca. 60%

Alternative für Windows:

Datapump Ordner komprimieren (Ersparnis: ca. 40%)

Datei nach Export manuell zippen (Ersparnis: ca. 70%)

Windows & UNIX

filesize=xG setzen und Dateien sofort nach Erzeugung komprimieren (z.B. mit File Watcher (ab 11.2))

Tipps & Tricks zur Performance Steigerung

Verwenden Sie mehrere parallele Export oder Import Vorgänge mit mehreren Dateien

```
DUMPFILERE=full_%U.dmp  
FILESIZE=1G  
PARALLEL=16
```

Legen Sie die Export/Import Dateien nicht auf die gleiche Platte, auf der der Tablespace liegt, von dem gelesen / auf den geschrieben wird.

Schreiben Sie die Export Datei auf eine Ramdisk, bzw. lesen Sie beim Import von dieser.

Schalten Sie ggf. die Archivierung und Flashback beim Import aus.

Schalten Sie beim Import die Statistikerzeugung aus, weil diese immer seriell abläuft

```
EXCLUDE=STATISTICS
```

Setzen Sie beim Import einen großen Sortierspeicher für die Indexerzeugung fest.

Kontaktadresse:

Marco Patzwahl
MuniQSoft GmbH
Grünwalder Weg 13a
D-82008 Unterhaching

Telefon: +49 (0) 89-6228 6789 40
Fax: +49 (0) 89-6228 6789 50
E-Mail: marco.patzwahl@muniqsoft.de
Internet: www.muniqsoft.de