ORACLE TRANSPORTABLE TABLESPACE KULLANIMI

BACKUP YÖNTEMLERİ KONUSUNDA ÖNBİLGİ:

Oracle'da yedek alınması için farklı yöntemler kullanılabilir. Veritabanını kapatarak (SHUTDOWN) dosyaların kopyalanmasıyla yedek alınabilir. Bu işlem Cold Backup olarak geçer. Ya da arşiv özelliği açıksa, Hot Backup yapılabilir. Diğer yöntemlerse RMAN'i kullanarak yedek alınması ve, import/export özelliklerinden yararlanmaktır.

Hepsinin kendine göre bir avantajı vardır.

i. Cold Backup işlemlerini yapmak basittir ve backup/recovery işlemleri çok hızlıdır. Ancak 7/24 açık kalması gereken sistemlerde veritabanını kapatma lüksü olmayabilir.
ii. Hot backup sıfır veri kaybını garanti eder ve yedek alınırken database'in kapanması gerekmez. Fakat DBA'ye ekstra iş düşer ve arşiv dosyaları ekstra disk ve işlemci masrafına neden olur.
iii. Import/Export işlemleri (Oracle10g ile birlikte datapump) database'in kapatılmasına gerek duymaz. Ancak backup ve recover işlemleri yavaş gerçekleşir. Ayrıca bir hata durumunda kaybedilen veri çok fazla olabilir. (Yoğun işlem yapılan bir veritabanında geceden alınan bir export, gün ortasında çöken bir veritabanı için başarıdan sayılmayacaktır.)
iv. RMAN (Recovery Manager) oldukça gelişmiş özellikler sunuyor. Hot backup özelliğini RMAN üzerinden kullanmak en iyi seçenek gibi gözükmekte.

Sayılan bu backup yöntemleri arasında sadece export ve import işlemleri, kullanılan işletim sisteminden bağımsız çalışır. İster cold backup alın, isterseniz arşivlerle hot backup şeklinde çalışın, kullandığınız işletim sisteminden, dilediğiniz her işletim sistemine geçiş yapmanız mümkün değildir. Big Endian-Little Endian denilen binary format biçimi bu geçişlere müsade etmemektedir. Big Endian'da en yüksek öneme sahip bit, en küçük adreste saklanır. Little Endian'da ise en düşük öneme sahip bit, en küçük adreste saklanır.

İŞLETİM SİSTEMLERİ ARASINDA GEÇİŞ:

Kullandığınız işletim sistemiyle, hedef işletim sisteminin endian yapısının aynı olup olmaması geçişlerin mümkün olup olmamasını belirler. Farklı işletim sistemleri eğer aynı endian yapısına sahipse, geçiş mümkün olabilir.

Aşağıda hangi işletim sisteminin hangi endian formatini kullanığını görebilirsiniz:

```
SQL> COLUMN PLATFORM_NAME FORMAT A32
SQL> SELECT * FROM V$TRANSPORTABLE_PLATFORM;
PLATFORM_ID PLATFORM_NAME                    ENDIAN_FORMAT
----------- -------------------------------- --------------
          1 Solaris[tm] OE (32-bit)          Big
          2 Solaris[tm] OE (64-bit)          Big
          7 Microsoft Windows IA (32-bit)    Little
         10 Linux IA (32-bit)                Little
          6 AIX-Based Systems (64-bit)       Big
          3 HP-UX (64-bit)                   Big
          5 HP Tru64 UNIX                    Little
          4 HP-UX IA (64-bit)                Big
         11 Linux IA (64-bit)                Little
         15 HP Open VMS                      Little
          8 Microsoft Windows IA (64-bit)    Little
          9 IBM zSeries Based Linux          Big
         13 Linux 64-bit for AMD             Little
         16 Apple Mac OS                     Big
         12 Microsoft Windows 64-bit for AMD Little
```

```
17 Solaris Operating System (x86)   Little
18 IBM Power Based Linux            Big
20 Solaris Operating System (AMD64) Little
19 HP IA Open VMS                   Little
```

Tabloya göre, Linux'tan Windows'a dosya taşınmasında bir sakınca yoktur. Ya da Solaris'te kurulu veritabanınızın yedeğini, AIX tabanlı sistemlerde recover etmeniz olasıdır. (Bunlar olasıdır ve Internet kaynaklarında yaptığını söyleyenler bulunuyor; ancak bizzat denemedim.) İşin problem çıkartan kısmı Little-endian formatıyla çalışan Linux'ta kurulu veritabanınızın yedeğini Big-Endian sistemiyle çalışan AIX sistemlere taşımaktadır. Bunu gerçekleştirmek için ***TRANSPORTABLE TABLESPACE*** özelliğinden yararlanmamız gerekir.

Transportable Tablespace Kullanımın Kısıtları

İşletim sistemleri arasında geçiş özelliğini kullanabilmemize karşın, Tranportable tablespace özelliğinin bir takım kısıtlamaları vardır:

- Kaynak ve hedef veritabanları aynı karakter setini kullanmalıdır.
- DENEME_TABLESPACE isimli bir tablespace'in kopyalanması söz konusu ise, hedef veritabanınında **KESİNLİKLE** bu isimde bir tablespace olmamalıdır.
- **SYSTEM, SYSAUX, TEMP, UNDOTBS1** tablespace'leri transfer edilemez.

Bunlar benim gözüme çarpanlar. Daha başka kullanım kısıtları da görünüyor; fakat yaptığım denemelerde beni etkilemediği için bunlardan bahsetmiyorum. Transporting Tablespaces Between Databases adresinde "***Limitations on Transportable Tablespace Use***" kısmına bakabilirsiniz.

Hazırlık Aşaması

Örneğimiz, Linux işletim sisteminden AIX işletim sistemine aktarılan bir veritabanıyla ilgilidir.

İşleme başlamadan önce AIX işletim sistemde tamamen boş bir veritabanı kurmanız uygun olacaktır. Bu sayede aynı isimde tablespace olma problemlerinden sakınabilirsiniz.

a. Kaynak veritabanında yapılacak işlemler:

Öncelikle hangi dosyaların, hangi konumdan taşınacağının belirlenmesi. Hangi tablespace'lerin taşınacağına aşağıdaki gibi karar verebilirsiniz:

```
SQL> COLUMN tablespace_name FORMAT A20
SQL> COLUMN name FORMAT A50
SQL> select tablespace_name, name from v$datafile_header;

TABLESPACE_NAME      NAME
-------------------- --------------------------------------------------
SYSTEM               /u01/app/oracle/oradata/TSH1/system01.dbf
UNDOTBS1             /u01/app/oracle/oradata/TSH1/undotbs01.dbf
SYSAUX               /u01/app/oracle/oradata/TSH1/sysaux01.dbf
USERS                /u01/app/oracle/oradata/TSH1/users01.dbf
DATA_TS              /u01/app/oracle/oradata/TSH1/data_ts_01.dbf
DATA_TS              /u01/app/oracle/oradata/TSH1/data_ts_02.dbf
DATA_TS              /u01/app/oracle/oradata/TSH1/data_ts_03.dbf
DATA_TS              /u01/app/oracle/oradata/TSH1/data_ts_04.dbf
```

```
DATA_TS              /u01/app/oracle/oradata/TSH1/data_ts_05.dbf

9 rows selected.
```

SYSTEM, SYSAUX, TEMP, UNDOTBS1 tablespace'lerinin transfer edilemeyeceğini daha önce söylemiştik. Bunun haricinde kalan her şeyi (yani DATA_TS ve USERS tablespace'lerini) transfer etmek istediğimizi varsayıyoruz.

Bir tablespace'in transfer edilebilmesi için '*self-contained*' olması, yani başka tablespace'lere bağımlı olmaması gerekir. Transfer edilecek veritabanlarının self-contained olup olmadığını aşağıdaki gibi test edebilirsiniz:

```
SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK(' USERS, DATA_TS', TRUE);
PL/SQL procedure successfully completed

SQL> SELECT * FROM TRANSPORT_SET_VIOLATIONS;
no rows selected
```

Sondaki select ifadesi boş döndüğü sürece (no rows selected) tablespace'leri taşımanızda bir sakınca yoktur. Fakat eğer sorgudan bir sonuç dönseydi, taşıma yapamayacağımız için ihtiyaç duyacağı tablespace'leri de dönüştürme işlemine dahil etmemiz gerekirdi.

Oracle kaynaklarında aşağıdaki işlemler geçmiyor. Fakat bunları yapmanızı tavsiye ederim. Böylece bağımlı olduğunuz tablespace'lerin read only konuma getirilmesi sizi etkilemez.

Geçiçi olarak kullanacağımız bir tablespace yaratıyoruz ve bunu kullanıcımıza atıyoruz:

```
SQL>  CREATE TABLESPACE DENEME_TABLESPACE DATAFILE
           '/tmp/DENEME_TABLEPACE.dbf'  SIZE 10M AUTOEXTEND ON NEXT 8K MAXSIZE 20M
          LOGGING
          ONLINE
          PERMANENT
          EXTENT MANAGEMENT LOCAL AUTOALLOCATE
          BLOCKSIZE 8K
          SEGMENT SPACE MANAGEMENT AUTO
          FLASHBACK ON;
Tablespace created.

SQL> ALTER USER d_ccebi
        DEFAULT TABLESPACE deneme_tablespace;
User altered.
```

Parolamı değiştiriyorum. Elbette bunu yapmanız gerekmiyor, mevcut parolanızla devam edebilirsiniz.

```
SQL> ALTER USER d_ccebi
        IDENTIFIED BY password;
User altered.
```

Dönüştüreceğimiz tablespace'leri read only mode'a çekiyoruz:

```
SQL> ALTER TABLESPACE USERS READ ONLY;
SQL> ALTER TABLESPACE DATA_TS READ ONLY;
```

Import işleminin geçerli olacağı klasörü yaratıp, Oracle'a tanıtıp, kullanıcımıza hak tanımlıyoruz:

```
SQL> ! mkdir /tmp/dpump_dir
SQL> CREATE OR REPLACE DIRECTORY DPUMP_DIR AS '/tmp/dpump_dir';
SQL> GRANT READ, WRITE ON DIRECTORY DPUMP_DIR TO D_CCEBI WITH GRANT OPTION;
```

Şimdi sıra, SQL Plus ortamından çıkıp dump dosyası oluşturmaya geldi. Bu dosya yapısal bilgi içerecek ve convert işlemi esnasında kullanılacak. (İşlem süresi yaklaşık 1.5 dakika sürüyor; database küçük olmasına rağmen, işlem süresi yine de kısa.)

```
oracle@hlodmpreprod:~ > expdp d_ccebi/password DUMPFILE=expdat.dmp DIRECTORY=DPUMP_DIR
TRANSPORT_TABLESPACES = USERS, DATA_TS
```

İşlem sonucunda /tmp/dpump_dir/expdat.dmp dosyasının oluştuğunu görebilirsiniz. Artık rman'e bağlanıp, convert işlemi yapabiliriz:

```
oracle@hlodmpreprod:~ > rman target /
...
connected to target database: TESTDB2 (DBID=136275210)
```

DBID'ye dikkat etmelisiniz. RMAN'e bağlanmadan önce select dbid from v$database ile, veritabanı ID'sini sorgulayabilir ve RMAN'in doğru database'e bağlanıp bağlanmadığından emin olabilirsiniz. **MUTLAKA DOĞRU DATABASE'E BAĞLANDIĞINIZDAN EMİN OLUN!**

RMAN'e bağlandıktan sonra aşağıdaki komutu çalıştırın:

```
RMAN> convert tablespace USERS, DATA_TS
       to platform 'AIX-Based Systems (64-bit)'
       db_file_name_convert '/u01/app/oracle/oradata/TSH1/',  '/tmp/dpump_dir/';
```

Bu komutla /u01/app/oracle/oradata/TSH1/ altında bulunan DBF dosyaları, /tmp/dpump_dir/ altına dönüştürülerek atılacaktır. Dönüştürme işleminde farklı komutlar kullanılabiliyor. Dosya adı farklılaştırılabiliyor vs... Aşağıdaki örneklerde dosya adı farklı biçime sokulmaktadır. Bunları kullanmanızı tavsiye etmiyorum. En ideali yukarıda vermiş olduğum komut olacaktır; bu komutla dosya adları aynı kalır.

```
RMAN> CONVERT TABLESPACE USERS, DATA_TS
       TO PLATFORM = 'AIX-Based Systems (64-bit)'
       FORMAT '/tmp/dpump_dir/%U';

RMAN> CONVERT TABLESPACE USERS, DATA_TS
       TO PLATFORM = 'AIX-Based Systems (64-bit)'
       FORMAT '/tmp/dpump_dir/%N_%f.dbf'
```

Format biçimleriyle ilgili bilgiyi Recovery Manager Reference (FormatSpec) adresinde bulabilirsiniz.

Convert tablespace işlemi bittikten sonra (yaklaşık 4 dakika sürüyor; yine hızlı sayılır) kaynak veritabanında yapılması gereken başka bir işlem kalmıyor. Aşağıdaki işlemleri uygulayarak database'i normal hâle döndürebilirsiniz.

Önce kullanıcımızın default tablespace'ini eskiden olduğu gibi USERS tablespace'i yapıyoruz:

```
SQL> ALTER USER d_ccebi
       DEFAULT TABLESPACE deneme_tablespace;
```

Dönüştüreceğimiz tablespace'leri read write mode'a geri çekiyoruz:

```
SQL> ALTER TABLESPACE USERS READ WRITE;
SQL> ALTER TABLESPACE DATA_TS READ WRITE;
```

Son olarak geçici olarak yarattığımız tablespace'i kaldıralım ve silelim:

```
SQL> DROP TABLESPACE DENEME_TABLESPACE;
SQL> ! rm /tmp/DENEME_TABLEPACE.dbf'
```

Dönüştürdüğünüz dosyaları dump dosyasıyla birlikte alıp (bizim örneğimizde /tmp/dpump_dir/ altındaki her şey), yeni database'in olduğu ortama aktarın.

b. Hedef veritabanında yapılacak işlemler:

Dosyaları aktardığımız yeri , /data2/ccebi_test/onliner0/hollanda olarak kabul edelim. Veritabanına bağlanıp, klasörü Oracle ortamında tanınlamamız ve D_CCEBI kullanıcısına yetki vermemiz gerekiyor:

```
SQL> CREATE OR REPLACE DIRECTORY  DPUMP_DIR AS '/data2/ccebi_test/onliner0/hollanda';
SQL> GRANT READ, WRITE ON DIRECTORY DPUMP_DIR TO D_CCEBI WITH GRANT OPTION;
```

Son aşamada data'yı veritabanına import ediyoruz. Aslında bu klasik anlamda bildiğimiz bir import işlemi değil, convert yapan bir import işlemi. Bunun için aşağıdaki komut çalıştırılıyor:

```
oracle3@test2: ~> impdp d_ccebi/password DUMPFILE=expdat.dmp DIRECTORY=dpump_dir
TRANSPORT_DATAFILES="'/data2/ccebi_test/onliner0/hollanda/data_ts_01.dbf','/data2/ccebi_test/onliner0/hollanda/data_ts_02.dbf',
'/data2/ccebi_test/onliner0/hollanda/data_ts_03.dbf','/data2/ccebi_test/onliner0/hollanda/data_ts_04.dbf','/data2/ccebi_test/onliner0/h
ollanda/data_ts_05.dbf', '/data2/ccebi_test/onliner0/hollanda/users01.dbf'"  REMAP_SCHEMA=D_AERTUNGA:anadolu
REMAP_SCHEMA=anadolu:anadolu
```

Yukarıdaki ifadede dosyaların bulunduğu konumları ve bu kimin schema'sından kime yazılacağını belirttik. (Tek tırnak ve çift tırnaklara dikkat etmek gerekiyor. İfadenin tek satır olduğunu da dikkat çekmek isterim.)

Yeni kurulan veritabanında ANADOLU isimli bir kullanıcının olması gerekiyor. Dikkat ettiyseniz, D_AERTUNGA ismini REMAP_SCHEMA'da girdiğimi görmüşsünüzdür. Girmediğiniz takdirde, import işlemi gerçekleşmeden size uyarı verilir. Böyle bir kullanıcı yok ve import edemiyorum hatası aldığınızda, o kullanıcıya ait bilgilerin başka bir kullanıcıya aktarılması gerektiğini bildirmeniz yeterli.

Yukarıdaki komutu girmenizle birlikte convert işlemini tamamlamış oluyorsunuz. Hızlandırmak için EXCLUDE=STATISTICS parametresini ekleyerek istatistik toplama işini sonraya bırakabilirsiniz.

Aktardığınız dosyaları kontrol edebilirsiniz:

```
SQL> COLUMN tablespace_name FORMAT A20
SQL> COLUMN name FORMAT A50
SQL> select tablespace_name, name from v$datafile_header where tablespace_name in
('DATA_TS','USERS');

TABLESPACE_NAME      NAME
-------------------- --------------------------------------------------
DATA_TS              /data2/ccebi_test/onliner0/hollanda/data_ts_05.dbf
DATA_TS              /data2/ccebi_test/onliner0/hollanda/data_ts_04.dbf
DATA_TS              /data2/ccebi_test/onliner0/hollanda/data_ts_03.dbf
DATA_TS              /data2/ccebi_test/onliner0/hollanda/data_ts_02.dbf
DATA_TS              /data2/ccebi_test/onliner0/hollanda/data_ts_01.dbf
USERS                /data2/ccebi_test/onliner0/hollanda/users01.dbf

6 rows selected.
```

KONUYA DAİR YORUMLAR

Aşırı zorunda kalmadıkça bu şekilde bir yöntem izlemek çok uygun değil. Tablespace'lerin bağımlılıklarını kontrol etseniz bile veritabanı rolleri nedeniyle hatalar alabiliyorsunuz.

Transportable tablespaces şeklinde yapılan çalışmalarda, işlem süresi normal sayılır. Yaptığım testlerde, import işlemini tamamlaması 4-5 dakika, istatistikleri güncellemesi ise 25 dakika kadar sürdü. Toplamda 30 dakikalık bir zaman gerekti. Elbette veritabanının boyutuna bağlı olarak bu süre azalıp-artacaktır. Fakat fikir vermesi açısından şunu söyleyebilirim; normal bir süreç izlenip, sadece veritabanının schema'sını klasik bir şekilde import ettiğimde, toplam işlem süresi 40-45 dakika kadar almaktaydı

Özetle, transportable tablespaces, kullanılabilir bir özellik. Ancak sonradan ortaya çıkabilecek ufak tefek hataları göze alarak bu işlemi gerçekleştirmelisiniz.