如何利用 RMAN 可传输表空间迁移数据库到不同字节序的平台 (Doc ID 1983639.1)

适用于:

Oracle Database - Enterprise Edition - 版本 10.1.0.2 到 12.1.0.1 [发行版 10.1 到 12.1]
Oracle Database Cloud Schema Service - 版本 N/A 和更高版本
Gen 1 Exadata Cloud at Customer (Oracle Exadata Database Cloud Machine) - 版本 N/A 和更高版本
Oracle Cloud Infrastructure - Database Service - 版本 N/A 和更高版本
Oracle Database Cloud Exadata Service - 版本 N/A 和更高版本
本文档所含信息适用于所有平台
********** 警告 **********

Document 1334152.1 Corrupt IOT when using Transportable Tablespace to HP from different OS
Document 13001379.8 Bug 13001379 - Datapump transport_tablespaces produces wrong dictionary metadata for some tables

目标

从 Oracle 数据库 10g 开始，你可以跨平台的传输表空间。这篇文档提供了一个逐步指导，来解释如何实现 ASM 数据文件和 OS 文件系统数据文件的传输表空间。

如果你的目标是迁移一个数据库到不同的字节序平台，如下的步骤概述了如何使用可传输表空间迁移一个数据库到一个新的平台：

1.- 在目标平台上创建一个新的，空的数据库。
2.- 从源库导入传输操作要求的对象到目标库。
3.- 从源库为所有的用户表空间导出可传输的元数据。
4.- 转移用户表空间的数据文件到目标系统。
5.- 使用 RMAN 转换数据文件到目标系统的字节序格式。
6.- 导入所有用户表空间的可传输元数据到目标数据库。

你也可以在源平台转换数据文件，转换完成后转移他们到目标平台。

MAA 技术简介"利用表空间传输实现平台迁移" 请参考：

http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-platformmigrationtts-129269.pdf

从 11.2.0.4，12c 之后，如果要转换到 Linux x86-64，那么参考如下文档：

Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup [1389592.1]

对于 12c 您也可以参考如下文档：

12c How Perform Cross-Platform Database Transport to different Endian Platform with RMAN Backup Sets Document 2013271.1

解决方案

支持的平台

请查询 V$TRANSPORTABLE_PLATFORM 来查看受支持的平台，并确定每个平台的字节序。

```
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
```

如果源平台和目标平台是不同的字节序，那么必须在源平台或者目标平台上做一个额外的步骤，来转换被传输的表空间到目标格式。如果它们是同样的字节序，那么不需要做转换，表空间可以像同平台那样传输。

传输表空间

1. *传输表空间前的准备工作*
    - 检查表空间是自包含的：

```
SQL> execute sys.dbms_tts.transport_set_check('TBS1,TBS2', true);
SQL> select * from sys.transport_set_violations;
```

> 注意：在表空间被传输之前，这些违反传输标准的问题必须被解决。

    - 要成功的运行传输表空间导出，表空间必须在 READ ONLY 模式：

```
SQL> ALTER TABLESPACE TBS1 READ ONLY;
SQL> ALTER TABLESPACE TBS2 READ ONLY;
```

2. *导出元数据*
    - 使用传统导出工具：

```
exp userid=\'sys/sys as sysdba\' file=tbs_exp.dmp log=tba_exp.log transport_tablespace=y tablespaces=TBS1,TBS2
```

    - 使用数据泵导出：
      首先创建数据泵使用的目录对象，例如：

```
CREATE OR REPLACE DIRECTORY dpump_dir AS '/tmp/subdir' ;
GRANT READ,WRITE ON DIRECTORY dpump_dir TO system;
```

      然后初始化数据泵导出：

```
expdp system/password DUMPFILE=expdat.dmp DIRECTORY=dpump_dir TRANSPORT_TABLESPACES = TBS1,TBS2
```

      如果你想要在执行一个传输表空间操作的同时进行严格的包含关系检查，那么使用 TRANSPORT_FULL_CHECK 参数。

```
expdp system/password DUMPFILE=expdat.dmp DIRECTORY = dpump_dir TRANSPORT_TABLESPACES= TBS1,TBS2
TRANSPORT_FULL_CHECK=Y
```

      如果被传输的表空间集不是自包含的，那么导出会失败。
3. *使用 V$TRANSPORTABLE_PLATFORM 来确定每个平台的字节序*，你可以在每个平台实例执行如下查询：

```
SELECT tp.platform_id,substr(d.PLATFORM_NAME,1,30), ENDIAN_FORMAT
FROM V$TRANSPORTABLE_PLATFORM tp, V$DATABASE d
WHERE tp.PLATFORM_NAME = d.PLATFORM_NAME;
```

   如果你发现字节序是不同的，那么传输表空间集时必须进行转换：

```
RMAN> convert tablespace TBS1 to platform="Linux IA (32-bit)" FORMAT '/tmp/%U';

RMAN> convert tablespace TBS2 to platform="Linux IA (32-bit)" FORMAT '/tmp/%U';
```

   然后复制数据文件和导出的文件到目标环境。
4. *导入可传输表空间*
    - 使用传统导入工具：

```
imp userid=\'sys/sys as sysdba\' file=tbs_exp.dmp log=tba_imp.log transport_tablespace=y
datafiles='/tmp/....','/tmp/...'
```

    - 使用数据泵：

```
CREATE OR REPLACE DIRECTORY dpump_dir AS '/tmp/subdir';
GRANT READ,WRITE ON DIRECTORY dpump_dir TO system;
```

      然后执行：

```
impdp system/password DUMPFILE=expdat.dmp DIRECTORY=dpump_dir TRANSPORT_DATAFILES='/tmp/....','/tmp/...'
REMAP_SCHEMA=(source:target) REMAP_SCHEMA=(source_sch2:target_schema_sch2)
```

      如果你想要改变传输的数据库对象的属主，可以使用 REMAP_SCHEMA。
5. *将表空间置于 read/write 模式：*

```
SQL> ALTER TABLESPACE TBS1 READ WRITE;
SQL> ALTER TABLESPACE TBS2 READ WRITE;
```

使用 DBMS_FILE_TRANSFER

你也可以是使用 DBMS_FILE_TRANSFER 来拷贝数据文件到另外一个主机。

> 从 12c 和 11.2.0.4 开始 DBMS_FILE_TRANSFER 默认的进行转换。若使用 DBMS_FILE_TRANSFER，当目标数据库收到一个来自不同字节序的平台的文件时，它对每一个块进行转换。作为可传输操作的一部分，在数据文件被移动到目标数据库后，不需 RMAN 转换，即可导入。

在低于 11.2.0.4 的版本上，对于 ASM 文件同样需要执行上面的步骤。但是如果字节序格式不同，那么你必须在转移文件后，使用 RMAN 转换。文件无法直接在不同平台的两个 ASM 实例间进行拷贝。

如下是一个使用范例：

```
RMAN> CONVERT DATAFILE
'/path/tbs_31.f',
'/path/tbs_32.f',
'/path/tbs_41.f'
TO PLATFORM="Solaris[tm] OE (32-bit)"
FROM PLATFORM="HP TRu64 UNIX"
DB_FILE_NAME_CONVERT= "/path_source/", "/path_dest/"
PARALLELISM=5;
```

相同的范例，但是这里显示目的地是一个 ASM 磁盘组：

```
RMAN> CONVERT DATAFILE
'/path/tbs_31.f',
'/path/tbs_32.f',
'/path/tbs_41.f'
TO PLATFORM="Solaris[tm] OE (32-bit)"
FROM PLATFORM="HP TRu64 UNIX"
DB_FILE_NAME_CONVERT="/path_source/", "+diskgroup"
PARALLELISM=5;
```

***** 警告*****

- **当使用可传输表空间（TTS）从 Solaris，Linux 或者 AIX 迁移到 HP/UX 时，索引组织表（IOT）可能损坏。**
  这是 BUG:9816640 带来的限制。
  当前针对这个问题没有补丁。索引组织表（IOT）需要在 TTS 之后进行重建。

  参考文档 1334152.1 Corrupt IOT when using Transportable Tablespace to HP from different OS。
- **当使用被 drop 掉的列，可能遇到这个 Bug:13001379 - Datapump transport_tablespaces produces wrong dictionary metadata for some tables can occur。文档 1440203.1 给出了这个警告的细节。**

使用 DBMS_FILE_TRANSFER 的已知问题

=> 未公开的 Bug 13636964 - ORA-19563 from RMAN convert on datafile copy transferred with DBMS_FILE_TRANSFER (Doc ID 13636964.8)
确认受影响的版本
11.2.0.3
问题在如下版本修复
12.1.0.1 (Base Release)
11.2.0.4 (Future Patch Set)

描述

使用 DBMS_FILE_TRANSFER 转移的文件在 RMAN convert 操作中失败。
例如：
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===
RMAN-03002: failure of conversion at target command at 01/24/2012 16:22:23
ORA-19563: cross-platform datafile header validation failed for file +RECO/tbs_9.tf

Rediscovery Notes:
如果 RMAN 转换一个使用 DBMS_FILE_TRANSFER 转移的文件失败，那么可能是由于这个 Bug。

规避方案：
使用 OS 工具转移文件。

=> Dbms_file_transfer Corrupts Dbf File When Copying between endians (Doc ID 1262965.1)

额外的资源

社区： Database Utilities

仍有其它问题吗？ 使用如上的社区来搜索相似的讨论，或者就此主题开启一个新的讨论。

可传输表空间使用的限制

1. 源库和目标库必须使用相同的字符集和国家字符集。
2. 如果目标库上已经有一个同名的表空间，无法进行传输。然而，你可以在传输之前，重命名要传输的表空间或者目标库上的表空间。

3. 若对象带有下层对象（例如物化视图）或者被包含的对象（例如分区表），则无法被传输。除非所有下层对象或者被包含的对象都在这个表空间集里。
    - 查看 Oracle Database Utilities 文档中的表"Objects Exported and Imported in Each Mode"，里面有几个对象类型在表空间模式中不被导出。
4. 如果表空间对象的所有者在目标库中不存在，则需要在开始可传输表空间导入之前，手动的创建用户名。

    - 如果你使用了 spatial 索引，那么：
        - 注意在 10gR1 和 10gR2 中，对于 spatial 索引，不支持跨不同字节序平台的 TTS 操作。这个限制在 11g 中取消了。
        - 在导出之前和传输之后，必须运行专门的 spatial 包，请参阅 Oracle Spatial 文档.
5. 从 Oracle Database 11gR1 开始，对于含有 XMLType 的表空间，必须使用数据泵来导出和导入表空间元数据。

    如下的查询返回了包含 XMLType 的表空间的列表：

    ```
    select distinct p.tablespace_name
    from dba_tablespaces p, dba_xml_tables x, dba_users u, all_all_tables t
    where t.table_name=x.table_name and
          t.tablespace_name=p.tablespace_name and
          x.owner=u.username;
    ```

    传输带有 XMLType 的表空间有如下限制
    a. 目标数据库必须安装 XML DB。
    b. XMLType 表引用的 schema 不能是 XML DB 标准 schema。
    c. XMLType 表引用的 schema 不能有循环依赖。
    d. XMLType 表上的任何行级别安全性都会在导入时丢失。
    e. 如果一个传输的 XMLType 表的 schema 不在目标数据库里，它会被导入并且注册。如果这个 schema 在目标数据库里已经存在了，就会返回一个错误，除非使用 ignore=y 选项。
6. 高级队列可传输表空间不支持带有多个容器的 8.0 兼容版高级队列。
7. 你无法传输 SYSTEM 表空间或者用户 SYS 拥有的对象。
8. 不透明类型（例如 RAW，BFILE 和 AnyTypes）可以被传输，但是他们不会在跨平台传输操作中被转换。他们的实际框架只有应用知道，所以应用必须在这些类型被移动到新的平台后处理字节序的问题。
9. 浮点数 BINARY_FLOAT 和 BINARY_DOUBLE 类型是可以传输的，但必须使用 Data Pump，不能使用原始的导出工具 EXP。
10. 其它更多的限制和要求,请查看以下文档： Document 1454872.1 - Transportable Tablespace (TTS) Restrictions and Limitations: Details, Reference, and Version Where Applicable

ASM 文件的可传输表空间导出/导入

- **使用 RMAN CONVERT**

  没有直接的方法将 ASM 文件作为可传输表空间导出/导入。但是，可以通过 RMAN 实现这个功能。

  请务必遵照如下步骤：
  1. *导出表空间前的准备。*

      - 检查表空间是自包含的：

        ```
        SQL>execute sys.dbms_tts.transport_set_check('TBS1,TBS2', true);
        SQL> select * from sys.transport_set_violations;
        ```

        > 注意：这些违反限制的结果必须在表空间传输前解决。

      - 要成功的进行可传输表空间导出，这些表空间必须处于 READ ONLY 模式。

        ```
        SQL> ALTER TABLESPACE TBS1 READ ONLY;
        SQL> ALTER TABLESPACE TBS2 READ ONLY;
        ```

  2. *导出元数据。*
      - 使用原始导出工具：

        ```
        exp userid=\'sys/sys as sysdba\' file=tbs_exp.dmp log=tba_exp.log transport_tablespace=y
        tablespaces=TBS1,TBS2
        ```

      - 使用数据泵导出：

        ```
        CREATE OR REPLACE DIRECTORY dpump_dir AS '/tmp/subdir';
        GRANT READ,WRITE ON DIRECTORY dpump_dir TO system;
        ```

        然后执行：

        ```
        expdp system/password DUMPFILE=expdat.dmp DIRECTORY=dpump_dir TRANSPORT_TABLESPACES = TBS1,TBS2
        ```

        如果你想要在执行可传输表空间操作同时进行严格的包容性检查，那么使用 TRANSPORT_FULL_CHECK 参数：

        ```
        expdp system/password DUMPFILE=expdat.dmp DIRECTORY = dpump_dir TRANSPORT_TABLESPACES= TBS1,TBS2
        TRANSPORT_FULL_CHECK=Y
        ```

    如果传输的表空间不是自包含的，那么导出会出错。
  3. *使用 V$TRANSPORTABLE_PLATFORM* 找到目标库准确的平台名。你可以在目标平台实例上执行如下的查询。

      ```
      SELECT tp.platform_id,substr(d.PLATFORM_NAME,2,30), ENDIAN_FORMAT
      FROM V$TRANSPORTABLE_PLATFORM tp, V$DATABASE d
      WHERE tp.PLATFORM_NAME = d.PLATFORM_NAME;
      ```

  4. *以目标平台的格式，从 ASM 文件生成一个 OS 文件。*

```
RMAN> CONVERT TABLESPACE TBS1
      TO PLATFORM 'HP-UX (64-bit)' FORMAT '/tmp/%U';
RMAN> CONVERT TABLESPACE TBS2
      TO PLATFORM 'HP-UX (64-bit)' FORMAT '/tmp/%U';
```

5. *拷贝生成的文件到目标服务器（如果跟源服务器不是同一台机器）。*
6. *导入可传输表空间。*
   - 使用原始的导入工具：

     ```
     imp userid=\'sys/sys as sysdba\' file=tbs_exp.dmp log=tba_imp.log transport_tablespace=y
     datafiles='/tmp/....','/tmp/...'
     ```

   - 使用数据泵导入：

     ```
     CREATE OR REPLACE DIRECTORY dpump_dir AS '/tmp/subdir';
     GRANT READ,WRITE ON DIRECTORY dpump_dir TO system;
     ```

     然后执行：

     ```
     impdp system/password DUMPFILE=expdat.dmp DIRECTORY=dpump_dir TRANSPORT_DATAFILES='/tmp/....','/tmp/...'
     REMAP_SCHEMA=(source:target) REMAP_SCHEMA=(source_sch2:target_schema_sch2)
     ```

     如果你想要改变传输的数据库对象的属主的话，可以使用 REMAP_SCHEMA 参数。
7. *将表空间放置在 read/write 模式。*

   ```
   SQL> ALTER TABLESPACE TBS1 READ WRITE;
   SQL> ALTER TABLESPACE TBS2 READ WRITE;
   ```

   如果你想要将数据文件从 ASM 环境传输到文件系统，那么操作到此结束。但如果你想要在两个 ASM 环境之间传输表空间，那么你要继续下面的操作。
8. *使用 rman 拷贝文件'/tmp/....dbf' 到 ASM 环境。*

   ```
   rman nocatalog target /
   RMAN> backup as copy datafile '/tmp/....dbf' format '+DGROUPA';
   ```

   这里 +DGROUPA 是 ASM 磁盘组名字。
9. *将数据文件交换到这个拷贝。*
   如果是 10g 数据库，首先要将数据文件离线：

   ```
   SQL> alter database datafile '/tmp/....dbf' offline;
   ```

   文件交换到这个拷贝：

   ```
   rman nocatalog target /
   RMAN> switch datafile '/tmp/....dbf' to copy;
   ```

   记下在 +DGROUPA 磁盘组中创建的拷贝的名字，例如，'+DGROUPA/SID/datafile/tts.270.5'。
10. *使文件重新在线，我们首先要 recover 它。*

    ```
    SQL> recover datafile '+DGROUPA/SID/datafile/tts.270.5';
    SQL> alter database datafile '+DGROUPA/SID/datafile/tts.270.5' online;
    ```

11. *检查数据文件确实已经是 ASM 环境的一部分，并且在线。*

    ```
    SQL> select name, status from v$datafile;
    ```

    输出应该是：

    ```
    +DGROUPA/SID/datafile/tts.270.5 ONLINE
    ```

- **使用DBMS_FILE_TRANSFER**

  你同样可以使用 DBMS_FILE_TRANSFER 来将数据文件从一个 ASM 磁盘组拷贝到另外一个，甚至到另外一个主机上。从 10gR2 开始你同样可以使用 DBMS_FILE_TRANSFER 来拷贝数据文件从 ASM 到文件系统，以及从文件系统到 ASM。

  PUT_FILE 过程读取一个本地文件或者 ASM 并且联系远端数据库来创建一个在远端文件系统的拷贝。被拷贝的文件是源文件，拷贝带来的新文件是目标文件。直到过程成功完成，目标文件都不会被关闭。

  语法：

  ```
  DBMS_FILE_TRANSFER.PUT_FILE(
     source_directory_object       IN  VARCHAR2,
     source_file_name              IN  VARCHAR2,
     destination_directory_object  IN  VARCHAR2,
     destination_file_name         IN  VARCHAR2,
     destination_database          IN  VARCHAR2);
  ```

  其中：
  - *source_directory_object*: 在本地源端拷贝的文件所在的目录对象。在源端，这个目录对象必须存在。
  - *source_file_name*: 从本地文件系统拷贝的文件的名字。这个文件必须存在于本地文件系统上 source_directory_object 所指定的目录里。
  - *destination_directory_object*: 这是在目标端文件所要放置的目录对象。这个目录对象必须存在于远端文件系统。
  - *destination_file_name*: 放在远端文件系统的文件的名字。在远端文件系统目标目录中必须没有重名的文件。
  - *destination_database*: 指向作为拷贝文件的目的地的远端数据库的数据库链接的名字。

如果我们想要使用 DBMS_FILE_TRANSFER.PUT_FILE 来从源端传输文件到目的地主机，步骤3，4，5做如下修改：

1. 在目标数据库主机创建一个目录，授权给本地用户。这是文件要在目标端放置的目录对象，必须在远端的文件系统存在。

```
CREATE OR REPLACE DIRECTORY target_dir AS '+DGROUPA';
GRANT WRITE ON DIRECTORY target_dir TO "USER";
```

2. 在源数据库主机创建一个目录。这是要拷贝的文件在本地源端所存在的目录对象。这个目录对象必须在源端存在。

```
CREATE OR REPLACE DIRECTORY source_dir AS '+DGROUPS/subdir';
GRANT READ,WRITE ON DIRECTORY source_dir TO "USER";
CREATE OR REPLACE DIRECTORY source_dir_1 AS '+DGROUPS/subdir/subdir_2';
```

3. 创建一个 dblink 连接到目标数据库主机：

```
CREATE DATABASE LINK DBS2 CONNECT TO 'user' IDENTIFIED BY 'password' USING 'target_connect';
```

   这里 target_connect 是目标数据库的连接字符串，USER 是我们将要用来转移数据文件的用户。

4. 连接到源实例。会用到如下项目：
   - dbs1: 到源数据库的连接字符串
   - dbs2: 到目标数据库的 dblink
   - a1.dat: 源数据库的文件名
   - a4.dat: 目标数据库的文件名

```
CONNECT user/password@dbs1

-- - put a1.dat to a4.dat (using dbs2 dblink)
-- - level 2 sub dir to parent dir
-- - user has read privs on source_dir_1 at dbs1 and write on target_dir
-- - in dbs2
BEGIN
    DBMS_FILE_TRANSFER.PUT_FILE('source_dir_1', 'a1.dat',
                                'target_dir', 'a4.dat', 'dbs2' );
END;
```

参考

BUG:13001379 - DATAPUMP TRANSPORT_TABLESPACES PRODUCES WRONG METADATA FOR SOME TABLES
http://docs.oracle.com/cd/E16655_01/appdev.121/e17602/d_ftran.htm#CHDEFEGG
NOTE:13001379.8 - Bug 13001379 - Datapump transport_tablespaces produces wrong dictionary metadata causing errors like ORA-7445 [qeaeMinmax] / ORA-1858 etc...
NOTE:1334152.1 - ALERT: Corrupt IOT when using Transportable Tablespace to HP from different OS
BUG:9816640 - ORA-600 [6200] ORA-600 [KDDUMMY_BLKCHK] IOT CORRUPTION CODE 6401 AFTER TTS
NOTE:1324000.1 - ASMCMD Copy to Convert Endianess using RMAN fails in 11g
NOTE:13636964.8 - Bug 13636964 - ORA-19563 from RMAN convert on datafile copy transferred with DBMS_FILE_TRANSFER

NOTE:1262965.1 - Dbms_file_transfer Corrupts Dbf File When Copying between endians
NOTE:1389592.1 - 11G - Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup
http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-platformmigrationtts-129269.pdf