在 12c 中使用备份集进行 RMAN ACTIVE DUPLICATE（新特性）（Doc ID 2226107.1）

文档内容

目标

解决方案

参考

适用于:

Oracle Database - Enterprise Edition - 版本 12.1.0.1 到 12.1.0.2 [发行版 12.1]
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
Oracle Database Exadata Express Cloud Service - 版本 N/A 和更高版本
本文档所含信息适用于所有平台

目标

这篇文章阐述了 RMAN ACTIVE DUPLICATE（从正在运行的数据库上复制数据）新特性，其中的 RMAN ACTIVE DUPLICATE 使用源库备份集来实现。

在 Oracle 数据库 12c 之前，ACTIVE DUPLICATE 进程使用生产数据库进程来通过网络发送镜像拷贝。由于复制过程直接正比于数据库大小，这会是一个非常耗时的动作。现在，在 12c 中，数据库复制过程被改善了，它使用备份集来取代镜像拷贝。结果是，数据库的大小相对的变小了，因为 RMAN 会跳过未使用块，提交了的 undo 块等。而且，你甚至可以使用压缩和 multi-section 选项来达到更快的复制。更进一步的，从目标端使用辅助的通道来将备份集拉过网络，取代了 12c 之前使用的推的方法。

在 Oracle 数据库 12c 中，有一个针对备份集的"拉"（或者还原）进程。首先建立一个到源库的连接。然后辅助实例以备份集的形式从源库获取需要的数据库文件。从辅助实例进行一次还原操作。因此，源库的资源利用的更少。

在目标实例和辅助实例上都需要 TNS 连接。

基于 DUPLICATE 语句，RMAN 动态的决定使用哪个进程（'推'或者'拉'），这确保目前已有的定制化脚本仍然能够运行。

• 当你指定 USING BACKUPSET，RMAN 使用拉'的方法。
• 当你在 DUPLICATE 命令前指定 SET ENCRYPTION，RMAN 自动的使用'拉'的方法并且创建备份集。发送到目的地的备份是加密的。
• SECTION SIZE 语句将数据文件划分成多个子部分，并且通过辅助实例上的多个通道并行的还原。要有效的利用并行，应分配更多的 AUXILIARY 通道。
• 使用 USING COMPRESSED BACKUPSET 语句确认，文件被以压缩备份集的方式传输。RMAN 会在创建备份集时进行未用块压缩，从而减少通过网络传输的备份集的大小。

解决方案

环境:

源库: <target_db_name>

辅助实例: <auxiliary_db_name>

RMAN> select con_id,name,open_mode from v$pdbs;

CON_ID NAME OPEN_MODE
---------- ----------------------------- ----------
2 PDB$SEED READ ONLY
3 <pdb_name>1 READ WRITE
4 <pdb_name>3 READ WRITE

```
准备辅助实例 <auxiliary_oracle_sid>:

-- 为辅助实例创建初始化参数文件

在初始化参数文件中设置初始化参数，要求的参数有:
例如:

*.compatible='12.1.0.0.0'
*.control_files='<auxiliary_path>/control01.ctl','/<auxiliary_fra>/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='<auxiliary_db_name>'
*.db_unique_name='<auxiliary_db_name>'
*.db_recovery_file_dest='/<path>/'
*.db_recovery_file_dest_size=4800m
*.diagnostic_dest='<path>/'
*.enable_pluggable_database=true
```

```
*.log_archive_format='%t_%s_%r.dbf'
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
log_file_name_convert='<primary_path>/','<auxiliary_path>'
```

- 为辅助实例创建一个 Oracle 密码文件。

Active 数据库复制必须有密码文件。在 Active 数据库复制中，它使用与目标数据库相同的 SYSDBA 密码利用密码文件直接连接到辅助实例。在这个例子中，
RMAN 复制源库密码文件到目标主机并且覆盖任何存在的辅助实例密码文件。

```
cp $ORACLE_HOME/dbs/orapw<target_sid> ORACLE_HOME/dbs/orapw<auxiliary_sid>
```

-- 在辅助位置创建必须的目录来放置数据文件和 $ADR_HOME 的 trace 文件。

-- 建立到辅助实例的 Oracle 网络连接。

如果你要从一个活动数据库使用备份集复制，辅助实例和目标数据库必须是可以通过 Oracle 网络使用的。

使用 sqlplus 开启辅助实例。

使用 SQL*Plus 连接到辅助实例，将它启动到 NOMOUNT 模式。

```
在辅助服务器上：
export ORACLE_SID=<auxiliary_sid>
export ORACLE_HOME=

sqlplus / as sysdba
SQL*Plus: Release 12.1.0.1.0 Production on Mon Mar 9 12:18:37 2015

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup nomount pfile='$ORACLE_HOME/dbs/pfile<oracle_sid>.ora';
ORACLE instance started.

Total System Global Area  242208768 bytes
Fixed Size                  2286944 bytes
Variable Size             184552096 bytes
Database Buffers           50331648 bytes
Redo Buffers                5038080 byt
```

- 在 listener.ora 和 tnsnames.ora 中创建必须的 Oracle 网络连接。请看例子：
在如下测试用例中，主库和备库运行在同一个主机 localhost.localdomain 上。

```
Listener.ora （在辅助实例主机）
SID_LIST_LISTENER =
  (SID_LIST =

    (SID_DESC =
      (GLOBAL_DBNAME = <auxiliary_db_name>)
      (ORACLE_HOME = <oracle_home>)
      (SID_NAME = <auxiliary_oracle_sid>)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = <target_db_name>)
      (ORACLE_HOME = <oracle_home>)
      (SID_NAME = <target_oracle_sid>)
    )
  )
LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
  )


 tnsname.ora （在目标和辅助实例主机）
 ----------------------------------------------
<target_service_name> =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = <target_service_name>)
    )
  )
<auxiliary_service_name> =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = <auxiliary_service_name>)
    )
  )
```

- 使用 tnsping 确认到目标和辅助实例的连接：
```
    % tnsping <target_service>
    % tnsping <auxiliary_service>
```

- 使用网络服务连接到目标和辅助数据库
```
RMAN target sys/<password>@<target_service> auxiliary sys/<password>@<auxiliary_service>
```

```
Recovery Manager: Release 12.1.0.1.0 - Production on Mon Mar 9 12:41:12 2015

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights reserved.

connected to target database: <db_name> (DBID=4165840403)
connected to auxiliary database: <db_name>(DBID=4165840403, not open)
```

下面的例子，我们会使用新特性'USING BACKUPSET' 来复制目标库 PRIM，但是排除掉 pluggable database <pdb_name>。

脚本：

```
RMAN> run{
2> allocate channel prmy1 type disk;
allocate channel prmy2 type disk;
allocate auxiliary channel aux1 type disk;
allocate auxiliary channel aux2 type disk;
allocate auxiliary channel aux3 type disk;
allocate auxiliary channel aux4 type disk;
DUPLICATE TARGET DATABASE TO '<auxiliary_db_name>'
PLUGGABLE DATABASE pdb1
FROM ACTIVE DATABASE
USING BACKUPSET
db_file_name_convert='<target_path>/','<auxiliary_path>';
```

复制日志：

```
Recovery Manager: Release 12.1.0.1.0 - Production on Fri Mar 6 00:18:55 2015

Copyright (c) 1982, 2013, Oracle and/or its affiliates. All rights reserved.

connected to target database: <db_name> (DBID=4165840403)
connected to auxiliary database: <auxiliary_db_name> (not mounted)

RMAN>

RMAN>

RMAN> run{
2> allocate channel prmy1 type disk;
allocate channel prmy2 type disk;
allocate auxiliary channel aux1 type disk;
allocate auxiliary channel aux2 type disk;
allocate auxiliary channel aux3 type disk;
allocate auxiliary channel aux4 type disk;
DUPLICATE TARGET DATABASE TO '<auxiliary_name>'
PLUGGABLE DATABASE <pdb_name>
FROM ACTIVE DATABASE
USING BACKUPSET
db_file_name_convert='<target_path>/','/<auxiliary_path>/';
}3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 13>

using target database control file instead of recovery catalog
allocated channel: prmy1
channel prmy1: SID=63 device type=DISK

allocated channel: prmy2
channel prmy2: SID=66 device type=DISK

allocated channel: aux1
channel aux1: SID=21 device type=DISK

allocated channel: aux2
channel aux2: SID=22 device type=DISK

allocated channel: aux3
channel aux3: SID=23 device type=DISK

allocated channel: aux4
channel aux4: SID=24 device type=DISK

Starting Duplicate Db at 06-MAR-15
current log archived

contents of Memory Script:
{
sql clone "create spfile from memory";
}
executing Memory Script

sql statement: create spfile from memory

contents of Memory Script:
{
shutdown clone immediate;
```

```
startup clone nomount;
}
executing Memory Script

Oracle instance shut down

connected to auxiliary database (not started)
Oracle instance started

Total System Global Area 242208768 bytes

Fixed Size 2286944 bytes
Variable Size 184552096 bytes
Database Buffers 50331648 bytes
Redo Buffers 5038080 bytes
allocated channel: aux1
channel aux1: SID=19 device type=DISK
allocated channel: aux2
channel aux2: SID=20 device type=DISK
allocated channel: aux3
channel aux3: SID=21 device type=DISK
allocated channel: aux4
channel aux4: SID=22 device type=DISK

contents of Memory Script:
{
sql clone "alter system set db_name =
''<db_name>'' comment=
''Modified by RMAN duplicate'' scope=spfile";
sql clone "alter system set db_unique_name =
''<db_name>'' comment=
''Modified by RMAN duplicate'' scope=spfile";
shutdown clone immediate;
startup clone force nomount
restore clone from service 'prim' primary controlfile;
alter clone database mount;
}
executing Memory Script

sql statement: alter system set db_name = ''<db_name>'' comment= ''Modified by RMAN duplicate'' scope=spfile

sql statement: alter system set db_unique_name = ''<db_unique_name>'' comment= ''Modified by RMAN duplicate''
scope=spfile

Oracle instance shut down

Oracle instance started

Total System Global Area 242208768 bytes

Fixed Size 2286944 bytes
Variable Size 184552096 bytes
Database Buffers 50331648 bytes
Redo Buffers 5038080 bytes
allocated channel: aux1
channel aux1: SID=19 device type=DISK
allocated channel: aux2
channel aux2: SID=20 device type=DISK
allocated channel: aux3
channel aux3: SID=21 device type=DISK
allocated channel: aux4
channel aux4: SID=22 device type=DISK

Starting restore at 06-MAR-15

channel aux1: starting datafile backup set restore
channel aux1: using network backup set from service prim
channel aux1: restoring control file
channel aux1: restore complete, elapsed time: 00:00:03
output file name=<path>/control01.ctl
output file name=<fra_path>/control02.ctl
Finished restore at 06-MAR-15

database mounted
Skipping pluggable database <pdb_name>
Automatically adding tablespace SYSTEM
Automatically adding tablespace SYSAUX
Automatically adding tablespace PDB$SEED:SYSTEM
Automatically adding tablespace PDB$SEED:SYSAUX
Automatically adding tablespace UNDOTBS1
Skipping tablespace USERS

contents of Memory Script:
{
set newname for datafile 1 to
"<path>/system01.dbf";
set newname for datafile 3 to
"<path>/sysaux01.dbf";
set newname for datafile 4 to
"<path>/undotbs01.dbf";
set newname for datafile 5 to
"<path>/pdbseed/system01.dbf";
set newname for datafile 7 to
"<path>/pdbseed/sysaux01.dbf";
set newname for datafile 8 to
```

```
"<path>/<pdb_name>/system01.dbf";
set newname for datafile 9 to
"<path>/<pdb_name>/sysaux01.dbf";
set newname for datafile 10 to
"<path>/<pdb_name>/<pdb_name>_users01.dbf";
restore
from service 'prim' clone database
skip forever tablespace "USERS",
"PDB3":"USERS",
"PDB3":"TEST",
"PDB3":"SYSTEM",
"PDB3":"SYSAUX" ;
sql 'alter system archive log current';
}
executing Memory Script

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

Starting restore at 06-MAR-15

channel aux1: starting datafile backup set restore
channel aux1: using network backup set from service <service_name>
channel aux1: specifying datafile(s) to restore from backup set
channel aux1: restoring datafile 00001 to <path>/system01.dbf
channel aux2: starting datafile backup set restore
channel aux2: using network backup set from service <service_name>
channel aux2: specifying datafile(s) to restore from backup set
channel aux2: restoring datafile 00003 to <path>/sysaux01.dbf
channel aux3: starting datafile backup set restore
channel aux3: using network backup set from service <service_name>
channel aux3: specifying datafile(s) to restore from backup set
channel aux3: restoring datafile 00004 to <path>/undotbs01.dbf
channel aux4: starting datafile backup set restore
channel aux4: using network backup set from service <service_name>
channel aux4: specifying datafile(s) to restore from backup set
channel aux4: restoring datafile 00005 to <path>/pdbseed/system01.dbf
channel aux3: restore complete, elapsed time: 00:00:15
channel aux3: starting datafile backup set restore
channel aux3: using network backup set from service <service_name>
channel aux3: specifying datafile(s) to restore from backup set
channel aux3: restoring datafile 00007 to <path>/pdbseed/sysaux01.dbf
channel aux4: restore complete, elapsed time: 00:01:01
channel aux4: starting datafile backup set restore
channel aux4: using network backup set from service <service_name>
channel aux4: specifying datafile(s) to restore from backup set
channel aux4: restoring datafile 00008 to <path>/<pdb_name>/system01.dbf
channel aux1: restore complete, elapsed time: 00:02:36
channel aux1: starting datafile backup set restore
channel aux1: using network backup set from service <service_name>
channel aux1: specifying datafile(s) to restore from backup set
channel aux1: restoring datafile 00009 to <path>/<pdb_name>/sysaux01.dbf
channel aux4: restore complete, elapsed time: 00:01:50
channel aux4: starting datafile backup set restore
channel aux4: using network backup set from service <service_name>
channel aux4: specifying datafile(s) to restore from backup set
channel aux4: restoring datafile 00010 to <path>/<pdb_name>/<pdb_name>_users01.dbf
channel aux4: restore complete, elapsed time: 00:00:07
channel aux3: restore complete, elapsed time: 00:03:21
channel aux2: restore complete, elapsed time: 00:04:06
channel aux1: restore complete, elapsed time: 00:02:40
Finished restore at 06-MAR-15

sql statement: alter system archive log current
current log archived

contents of Memory Script:
{
restore clone force from service '<service_name>'
archivelog from scn 1859981;
switch clone datafile all;
}
executing Memory Script

Starting restore at 06-MAR-15

channel aux1: starting archived log restore to default destination
channel aux1: using network backup set from service prim
channel aux1: restoring archived log
archived log thread=1 sequence=23
channel aux2: starting archived log restore to default destination
channel aux2: using network backup set from service prim
```

```
channel aux2: restoring archived log
archived log thread=1 sequence=24
channel aux1: restore complete, elapsed time: 00:00:01
channel aux2: restore complete, elapsed time: 00:00:01
Finished restore at 06-MAR-15

datafile 1 switched to datafile copy
input datafile copy RECID=12 STAMP=873592807 file name=<path>/system01.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=13 STAMP=873592807 file name=<path>/sysaux01.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=14 STAMP=873592807 file name=<path>/undotbs01.dbf
datafile 5 switched to datafile copy
input datafile copy RECID=15 STAMP=873592808 file name=<path>/pdbseed/system01.dbf
datafile 7 switched to datafile copy
input datafile copy RECID=16 STAMP=873592808 file name=<path>/pdbseed/sysaux01.dbf
datafile 8 switched to datafile copy
input datafile copy RECID=17 STAMP=873592808 file name=<path>/<pdb_name>/system01.dbf
datafile 9 switched to datafile copy
input datafile copy RECID=18 STAMP=873592808 file name=<path>/<pdb_name>/sysaux01.dbf
datafile 10 switched to datafile copy
input datafile copy RECID=19 STAMP=873592808 file name=<path>/<pdb_name>/<pdb_name>_users01.dbf

contents of Memory Script:
{
set until scn 1860344;
recover
clone database
skip forever tablespace "USERS",
"PDB3":"USERS",
"PDB3":"TEST",
"PDB3":"SYSTEM",
"PDB3":"SYSAUX" delete archivelog
;
}
executing Memory Script

executing command: SET until clause

Starting recover at 06-MAR-15

Executing: alter database datafile 6 offline drop
Executing: alter database datafile 18 offline drop
Executing: alter database datafile 19 offline drop
Executing: alter database datafile 16 offline drop
Executing: alter database datafile 17 offline drop
starting media recovery

archived log for thread 1 with sequence 23 is already on disk as file
<path>/archivelog/2015_03_06/o1_mf_1_23_bhkbjh1o_.arc
archived log for thread 1 with sequence 24 is already on disk as file
<path>/archivelog/2015_03_06/o1_mf_1_24_bhkbjh28_.arc
archived log file name=<path>/archivelog/2015_03_06/o1_mf_1_23_bhkbjh1o_.arc thread=1 sequence=23
archived log file name=<path>/archivelog/2015_03_06/o1_mf_1_24_bhkbjh28_.arc thread=1 sequence=24
media recovery complete, elapsed time: 00:00:01
Finished recover at 06-MAR-15
Oracle instance started

Total System Global Area 242208768 bytes

Fixed Size 2286944 bytes
Variable Size 184552096 bytes
Database Buffers 50331648 bytes
Redo Buffers 5038080 bytes

contents of Memory Script:
{
sql clone "alter system set db_name =
''<db_name>'' comment=
''Reset to original value by RMAN'' scope=spfile";
sql clone "alter system reset db_unique_name scope=spfile";
}
executing Memory Script

sql statement: alter system set db_name = ''PRIMDUP'' comment= ''Reset to original value by RMAN'' scope=spfile

sql statement: alter system reset db_unique_name scope=spfile
Oracle instance started

Total System Global Area 242208768 bytes

Fixed Size 2286944 bytes
Variable Size 184552096 bytes
Database Buffers 50331648 bytes
Redo Buffers 5038080 bytes
sql statement: CREATE CONTROLFILE REUSE SET DATABASE "<db_name>" RESETLOGS ARCHIVELOG
MAXLOGFILES 16
MAXLOGMEMBERS 3
MAXDATAFILES 1024
MAXINSTANCES 8
MAXLOGHISTORY 292
LOGFILE
GROUP 1 ( '<path>/redo01.log' ) SIZE 50 M REUSE,
GROUP 2 ( '<path>/redo02.log' ) SIZE 50 M REUSE,
GROUP 3 ( '<path>/redo03.log' ) SIZE 50 M REUSE
```

```
DATAFILE
'<path>/system01.dbf',
'/<path>/pdbseed/system01.dbf',
'/<path>/<pdb_name>/system01.dbf'
CHARACTER SET WE8MSWIN1252


contents of Memory Script:
{
set newname for tempfile 1 to
"<path>/temp01.dbf";
set newname for tempfile 2 to
"<path>/pdbseed/pdbseed_temp01.dbf";
set newname for tempfile 3 to
"<path>/<pdb_name>/temp01.dbf";
switch clone tempfile all;
catalog clone datafilecopy "<path>/sysaux01.dbf",
"/<path>/undotbs01.dbf",
"/<path>/pdbseed/sysaux01.dbf",
"/<path>/<pdb_name>/sysaux01.dbf",
"/<path>/<pdb_name>/<pdb_name>_users01.dbf";
switch clone datafile all;
}
executing Memory Script

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

renamed tempfile 1 to <path>/temp01.dbf in control file
renamed tempfile 2 to <path>/pdbseed/pdbseed_temp01.dbf in control file
renamed tempfile 3 to <path>/<pdb_name>/temp01.dbf in control file

cataloged datafile copy
datafile copy file name=<path>/sysaux01.dbf RECID=1 STAMP=873592829
cataloged datafile copy
datafile copy file name=<path>/undotbs01.dbf RECID=2 STAMP=873592829
cataloged datafile copy
datafile copy file name=<path>/pdbseed/sysaux01.dbf RECID=3 STAMP=873592829
cataloged datafile copy
datafile copy file name=<path>/<pdb_name>/sysaux01.dbf RECID=4 STAMP=873592829
cataloged datafile copy
datafile copy file name=<path>/<pdb_name>/<pdb_name>_users01.dbf RECID=5 STAMP=873592829

datafile 3 switched to datafile copy
input datafile copy RECID=1 STAMP=873592829 file name=<path>/sysaux01.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=2 STAMP=873592829 file name=<path>/undotbs01.dbf
datafile 7 switched to datafile copy
input datafile copy RECID=3 STAMP=873592829 file name=<path>/pdbseed/sysaux01.dbf
datafile 9 switched to datafile copy
input datafile copy RECID=4 STAMP=873592829 file name=<path>/<pdb_name>/sysaux01.dbf
datafile 10 switched to datafile copy
input datafile copy RECID=5 STAMP=873592829 file name=<path>/<pdb_name>/<pdb_name>_users01.dbf

contents of Memory Script:
{
Alter clone database open resetlogs;
}
executing Memory Script

database opened
Executing: drop pluggable database "<pdb_name>"

contents of Memory Script:
{
sql clone "alter pluggable database all open";
}
executing Memory Script

sql statement: alter pluggable database all open
Dropping offline and skipped tablespaces
Executing: alter database default tablespace system
Executing: drop tablespace "USERS" including contents cascade constraints
Cannot remove created server parameter file
Finished Duplicate Db at 06-MAR-15
released channel: prmy1
released channel: prmy2
```

上面的操作可能会遇到如下错误：

```
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of Duplicate Db command at 03/05/2015 22:57:52
```

RMAN-05501: aborting duplication of target database
RMAN-03015: error occurred in stored script Memory Script
RMAN-03009: failure of IRESTORE command on aux1 channel at 03/05/2015 22:57:52
RMAN-10032: unhandled exception during execution of job step 1:
ORA-06512: at line 140
ORA-19583: conversation terminated due to error
ORA-19849: error while reading backup piece from service prim
ORA-12582: TNS:invalid operation

此问题是由于 'Unpublished' Bug 18650329 : OGG:CDB: RMAN ACTIVE FAILS WITH ORA-12582: TNS:INVALID OPERATION

Bug 在 12.1.0.2上 修复。

绕开方案是：

在 sqlnet.ora 文件中添加 SQLNET.USE_ZERO_COPY_IO=0

参考

Didn't find what you are looking for?