

Oracle 高级复制冲突解决机制的研究

——使用 dbms_rectifier_diff 及手动解决 Oracle 高级复制中的冲突

本文作者: eygle (eygle.com@gmail.com)

摘要: 本文简要探讨 dbms_rectifier_diff 包的机制及如何手动解决 Oracle 高级复制中的冲突。

关于 dbms_rectifier_diff 的使用请参考:

http://www.eygle.com/archives/2005/01/eoadbms_rectifi.html

一、DIFFERENCES 过程的内部操作

实际上 Oracle 的 dbms_rectifier_diff.DIFFERENCES 过程，内部操作就是执行两个 minus 操作把两边的差异记录下来，作为冲突解决的数据。

这部分后台操作可以通过跟踪 Oracle 进程得到:

```
SQL> alter session set events '10046 trace name context forever,level 12';
```

```
Session altered.
```

```
Elapsed: 00:00:00.02
```

```
SQL> begin dbms_rectifier_diff.DIFFERENCES(  
 2  SNAME1                =>'HAWA',  
 3  ONAME1                =>'TEST',  
 4  REFERENCE_SITE       =>'AVATAR.COOLYOUNG.COM.CN',  
 4  SNAME2                =>'HAWA',  
 6  ONAME2                =>'TEST',  
 7  COMPARISON_SITE      =>'AUTHAA.COOLYOUNG.COM.CN',  
 8  WHERE_CLAUSE         =>NULL,  
 9  COLUMN_LIST          =>NULL,  
10  MISSING_ROWS_SNAME   =>'HAWA',  
11  MISSING_ROWS_ONAME1  =>'MISSING_ROWS_TEST',  
12  MISSING_ROWS_ONAME2  =>'MISSING_LOCATION_TEST',  
13  MISSING_ROWS_SITE   =>'AVATAR.COOLYOUNG.COM.CN',  
14  MAX_MISSING          =>500,  
15  COMMIT_ROWS         =>100  
16 );  
17 end;  
18 /
```

```
PL/SQL procedure successfully completed.
```

```
Elapsed: 00:00:01.97
```

```
SQL> alter session set events '10046 trace name context off';
```

从跟踪文件中我们可以清晰的看到(注意你所定义的所有参数在此都会有所体现):

1.首先是一个正向 Minus

```
DECLARE
    row_count      BINARY_INTEGER := 0;
    missing_rows   BINARY_INTEGER := 0;
    arowid         ROWID;

    CURSOR c
    IS
        SELECT "DATLOGONTIME", "NUMGENDER", "NUMSTATUS", "NUMUSERID", "VC2IP",
               "VC2USERNAME"
           FROM "HAWA"."TEST"
        MINUS
        SELECT "DATLOGONTIME", "NUMGENDER", "NUMSTATUS", "NUMUSERID", "VC2IP",
               "VC2USERNAME"
           FROM "HAWA"."TEST"@authaa.coolyoung.com.cn;
BEGIN
    FOR r IN c
    LOOP
        missing_rows := missing_rows + 1;

        IF missing_rows > 500
        THEN
            COMMIT;
            EXIT;
        END IF;

        INSERT INTO "HAWA"."MISSING_ROWS_TEST"
            ("DATLOGONTIME", "NUMGENDER", "NUMSTATUS",
             "NUMUSERID", "VC2IP", "VC2USERNAME"
            )
        VALUES (r."DATLOGONTIME", r."NUMGENDER", r."NUMSTATUS",
                r."NUMUSERID", r."VC2IP", r."VC2USERNAME"
                );

        SELECT ROWID
           INTO arowid
```

```
FROM "HAWA"."MISSING_ROWS_TEST"
WHERE ( datlogontime = r."DATLOGONTIME"
        OR (datlogontime IS NULL AND r."DATLOGONTIME" IS NULL)
      )
AND ( numgender = r."NUMGENDER"
      OR (numgender IS NULL AND r."NUMGENDER" IS NULL)
    )
AND ( numstatus = r."NUMSTATUS"
      OR (numstatus IS NULL AND r."NUMSTATUS" IS NULL)
    )
AND (numuserid = r."NUMUSERID")
AND (vc2ip = r."VC2IP" OR (vc2ip IS NULL AND r."VC2IP" IS NULL))
AND ( vc2username = r."VC2USERNAME"
      OR (vc2username IS NULL AND r."VC2USERNAME" IS NULL)
    );

INSERT INTO "HAWA"."MISSING_LOCATION_TEST"
      (present, absent, r_id
      )
VALUES ('AVATAR.COOLYOUNG.COM.CN', 'AUTHAA.COOLYOUNG.COM.CN',
        arowid
      );

row_count := row_count + 1;

IF row_count >= 100
THEN
  COMMIT;
  row_count := 0;
END IF;
END LOOP;

COMMIT;
END;
```

2.其次是一个反向 Minus

```
DECLARE
  row_count      BINARY_INTEGER := 0;
  missing_rows   BINARY_INTEGER := 0;
  arowid         ROWID;

  CURSOR c
```

```
IS
SELECT "DATLOGONTIME", "NUMGENDER", "NUMSTATUS", "NUMUSERID", "VC2IP",
      "VC2USERNAME"
FROM "HAWA"."TEST"@authaa.coolyoung.com.cn
MINUS
SELECT "DATLOGONTIME", "NUMGENDER", "NUMSTATUS", "NUMUSERID", "VC2IP",
      "VC2USERNAME"
FROM "HAWA"."TEST";
BEGIN
FOR r IN c
LOOP
missing_rows := missing_rows + 1;

IF missing_rows > 500
THEN
COMMIT;
EXIT;
END IF;

INSERT INTO "HAWA"."MISSING_ROWS_TEST"
      ("DATLOGONTIME", "NUMGENDER", "NUMSTATUS",
      "NUMUSERID", "VC2IP", "VC2USERNAME"
      )
VALUES (r."DATLOGONTIME", r."NUMGENDER", r."NUMSTATUS",
      r."NUMUSERID", r."VC2IP", r."VC2USERNAME"
      );

SELECT ROWID
INTO arowid
FROM "HAWA"."MISSING_ROWS_TEST"
WHERE ( datlogontime = r."DATLOGONTIME"
      OR (datlogontime IS NULL AND r."DATLOGONTIME" IS NULL)
      )
AND ( numgender = r."NUMGENDER"
      OR (numgender IS NULL AND r."NUMGENDER" IS NULL)
      )
AND ( numstatus = r."NUMSTATUS"
      OR (numstatus IS NULL AND r."NUMSTATUS" IS NULL)
      )
AND (numuserid = r."NUMUSERID")
AND (vc2ip = r."VC2IP" OR (vc2ip IS NULL AND r."VC2IP" IS NULL))
AND ( vc2username = r."VC2USERNAME"
```

```
        OR (vc2username IS NULL AND r."VC2USERNAME" IS NULL)
    );

    INSERT INTO "HAWA"."MISSING_LOCATION_TEST"
        (present, absent,
         r_id
        )
    VALUES ('AUTHAA.COOLYOUNG.COM.CN', 'AVATAR.COOLYOUNG.COM.CN',
            arowid
    );

    row_count := row_count + 1;

    IF row_count >= 100
    THEN
        COMMIT;
        row_count := 0;
    END IF;
END LOOP;

COMMIT;

END;
```

经过这两个步骤的操作，Oracle 定位了冲突数据。

二、冲突解决

可是注意，如果在解决这个问题时你没有挂起复制，Oracle 得到的数据可能是存在问题的。

而且，如果你不指定 column list,那么两边的数据可能会因为某些特殊字段(如时间字段)的特殊处理而存在差异。

那么这时候手工介入不可避免。

我们首先先把两个重要参数的用法说明一下。

一个是 WHERE_CLAUSE，另外一个为 COLUMN_LIST。

WHERE_CLAUSE 用于限定进行差异比较的范围，这可以极大的缩减结果集的数量，使用索引加快访问速度等。

比如我这里使用 NUMGENDER=1，只比较性别为"女"这一部分用户数据。

COLUMN_LIST 用于限定比较字段，如果你能通过某个字段，如主键等确定数据差异，那么你完全可以只比较单个字段。

而且显然可以轻易通过全索引扫描来完成比较，加快比较速度。

我这里使用 NUMUSERID，用户 ID 来比较。

但是注意，这样的比较结果中将只包含 NUMUSERID 信息，当然我们可以轻易通过 NUMUSERID 和原表的比较补全 MISSING_ROWS_TEST 表的信息。

```
begin dbms_rectifier_diff.DIFFERENCES(  
  SNAME1                =>'HAWA' ,  
  ONAME1                =>'TEST' ,  
  REFERENCE_SITE        =>'AVATAR.COOLYOUNG.COM.CN' ,  
  SNAME2                =>'HAWA' ,  
  ONAME2                =>'TEST' ,  
  COMPARISON_SITE       =>'AUTHAA.COOLYOUNG.COM.CN' ,  
  WHERE_CLAUSE          =>'NUMGENDER=1' ,  
  COLUMN_LIST           =>'NUMUSERID' ,  
  MISSING_ROWS_SNAME    =>'HAWA' ,  
  MISSING_ROWS_ONAME1   =>'MISSING_ROWS_TEST' ,  
  MISSING_ROWS_ONAME2   =>'MISSING_LOCATION_TEST' ,  
  MISSING_ROWS_SITE     =>'AVATAR.COOLYOUNG.COM.CN' ,  
  MAX_MISSING           =>500 ,  
  COMMIT_ROWS           =>100  
);  
end;  
/
```

这段代码供参考。

Ok，我们继续前面的讨论。

我们提到，如果存在差异，通常需要手工介入。

清楚了 DIFFERENCES 的原理，实际上我们完全可以手工来完成这个过程。

以下是我的手工操作步骤，目的是为了准确性及减轻数据库压力：

1.首先创建一个 ID 差异表

这个表不是必须的，这里是为了清晰

```
SQL> create table hawa.prof as select NUMUSERID from hawa.hw_user where 1=0;  
  
Table created.  
  
Elapsed: 00:00:00.16
```

2.根据主键找到差异记录

注意这里取决于你的数据库产生差异的原因，我的差异由于初始数据不同步，即 A 全包含 B 并且，A>B。

```
SQL> insert into hawa.prof
  2  select * from
  3  (
  4  select NUMUSERID from hawa.HW_USERPROFILE
  5  minus
  6  select NUMUSERID from hawa.HW_USERPROFILE@authaa)
  7  /

263 rows created.

Elapsed: 00:00:32.49
```

3.创建记录表

```
SQL> create table hawa.missing_rows_hw_userprofile
  2  as
  3  select * from hawa.hw_userprofile where 1=0;

Table created.

Elapsed: 00:00:00.12
```

4.创建位置(Location)表

注意这里 Oracle 需要记录缺失方向，和具体记录的 ROWID，这个 ROWID 来自 missing_rows_hw_userprofile。

```
SQL> create table hawa.MISSING_LOC_hw_userprofile (
  2  present VARCHAR2(128),
  3  absent VARCHAR2(128),
  4  r_id ROWID);

Table created.

Elapsed: 00:00:00.04
```

5.根据差异信息查询到完整信息

```
SQL> insert into hawa.missing_rows_hw_userprofile
  2  select * from hawa.hw_userprofile where NUMUSERID in
  3  (select * from hawa.prof);
```

```
263 rows created.

Elapsed: 00:00:00.06
SQL> commit;

Commit complete.

Elapsed: 00:00:00.02
```

6.构造位置信息

注意这里的方向信息及 ROWID 信息。

```
SQL> insert into hawa.MISSING_LOC_hw_userprofile
  2      select      'AVATAR.COOLYOUNG.COM.CN', 'AUTHAA.COOLYOUNG.COM.CN',rowid      from
hawa.missing_rows_hw_userprofile;

263 rows created.

Elapsed: 00:00:00.00
SQL> commit;

Commit complete.

Elapsed: 00:00:00.06
```

7.纠正数据冲突

```
SQL> BEGIN DBMS_RECTIFIER_DIFF.RECTIFY(
  2  SNAME1              =>'HAWA' ,
  3  ONAME1              =>'HW_USERPROFILE' ,
  4  REFERENCE_SITE     =>'AVATAR.COOLYOUNG.COM.CN' ,
  5  SNAME2              =>'HAWA' ,
  6  ONAME2              =>'HW_USERPROFILE' ,
  7  COMPARISON_SITE    =>'AUTHAA.COOLYOUNG.COM.CN' ,
  8  COLUMN_LIST        =>NULL ,
  9  MISSING_ROWS_SNAME =>'HAWA' ,
 10  MISSING_ROWS_ONAME1=>'MISSING_ROWS_HW_USERPROFILE' ,
 11  MISSING_ROWS_ONAME2=>'MISSING_LOC_HW_USERPROFILE' ,
 12  MISSING_ROWS_SITE  =>'AVATAR.COOLYOUNG.COM.CN' ,
 13  COMMIT_ROWS        =>100
```

```
14 );  
15 END;  
16 /
```

PL/SQL procedure successfully completed.

Elapsed: 00:00:03.53

8.验证结果

```
SQL> select count(*) from hawa.HW_USERPROFILE;
```

```
  COUNT(*)  
-----  
  1746300
```

Elapsed: 00:00:02.22

```
SQL> select count(*) from hawa.HW_USERPROFILE@authaa;
```

```
  COUNT(*)  
-----  
  1746300
```

Elapsed: 00:00:00.21

```
SQL> select count(*) from hawa.HW_USERPROFILE;
```

```
  COUNT(*)  
-----  
  1746300
```

Elapsed: 00:00:00.59

```
SQL>select count(*) from hawa.HW_USERPROFILE@authaa;
```

```
  COUNT(*)  
-----  
  1746300
```

Elapsed: 00:00:00.20

```
SQL> select NUMUSERID from hawa.HW_USERPROFILE
```

```
2 minus
3 select NUMUSERID from hawa.HW_USERPROFILE@authaa ;

no rows selected

Elapsed: 00:00:23.51
SQL>
```

作者简介：



盖国强，网名 eygle

曾任 ITPUB MS 版版主,现任 itpub Oracle 管理版版主.

曾任职于某国家大型企业,服务于烟草行业,开发过基于 Oracle 数据库的大型 ERP 系统,属国家信息产业部重点工程.同时负责 Oracle 数据库管理及优化,并为多家烟草企业提供 Oracle 数据库管理、优化及技术支持.

目前任职于北京某电信增值业务系统提供商企业,首席 DBA,负责数据库业务.管理全国 30 多个数据库系统。项目经验丰富,曾设计规划及支持中国联通增值业务等大型数据库系统.

实践经验丰富，长于数据库诊断、性能调整与 SQL 优化等。对于 Oracle 内部技术具有深入研究。

高级培训讲师,培训经验丰富,曾主讲 itpub dba 培训及 itpub 高级性能调整等主要课程.《Oracle 数据库 DBA 专题技术精粹》一书的主编及主要作者.

你可以在<http://www.eygle.com>上找到关于作者的更多信息.