

ParnassusData is a software company

oracle 10R2 streams 实验

Parnassus

诗檀

诗檀软件 www.parnassusdata.com

诗檀软件Oracle技术团队，国内最专业的Oracle数据库服务, <http://www.parnassusdata.com>

工程师：	郭兆伟
报告生成日期：	2015年7月20日
更新日期：	2015年7月20日



文档控制

此文档仅供诗檀软件内部审阅，不得向与此无关的个人或机构传阅或复制。

变更记录

日期	作者及更新人	版本号	变更信息

审阅人

版本号	审阅人	职位	相关评论
1.0			

审批人

版本号	批准人	日期	相关评论

文档分发

分发号	文档名	分发位置
1	SHCH-1	ASANA

目录

文档控制.....	2
变更记录.....	2
审阅人.....	2
审批人.....	2
文档分发.....	2
目录.....	3
1. 实验环境.....	4
2.实验及其结果.....	4
3.stream 环境设置.....	4
3.1 确定数据库参数配置：.....	4
3.2 确认数据库是归档模式：.....	4
3.3 开启 force logging.....	4
3.4 创建 stream 用户和相关表空间：.....	4
3.5 赋权.....	5
3.6 创建 DB link.....	5
3.7 在源端和目标端分别创建队列.....	5
3.8 创建进程.....	6
3.9 初始化数据：.....	7
3.10 创建传播进程.....	8
4 stream 启停.....	9
4.1 启动.....	9
4.2 停止.....	10
5 实际测试.....	10
6.其他信息.....	12
6.1 日志信息.....	12
6.2 参考博客.....	15
5.总结.....	16
6.其他问题.....	16
6.1 未解决的问题.....	16
已解决的问题.....	16

1. 实验环境

虚拟机环境 rhel 5.6_64 oracle 10.2.0.4

内核参数调整：

```
cat <<EOF >>/etc/sysctl.conf
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
EOF
sysctl -p
```

2. 实验及其结果

完成了双节点的双向复制

3. stream 环境设置

3.1 确定数据库参数配置：

```
alter system set "_job_queue_interval"=1 SCOPE=SPFILE;
alter system set GLOBAL_NAMES=true SCOPE=SPFILE;
alter system set JOB_QUEUE_PROCESSES=4 SCOPE=SPFILE;
alter system set UNDO_RETENTION=3600 scope=spfile;
alter system set log_archive_dest_1='location=/home/oracle/arch' scope=spfile;
startup force
```

3.2 确认数据库是归档模式：

```
SQL> archive log list
Database log mode                Archive Mode
Automatic archival              Enabled
Archive destination              /home/oracle/arch
Oldest online log sequence      3
Next log sequence to archive    5
Current log sequence            5
```

3.3 开启 force logging

```
alter database force logging;
```

3.4 创建 stream 用户和相关表空间：

```
CREATE TABLESPACE strmadmin DATAFILE
```

```
'$ORACLE_BASE/$ORACLE_SID/strmadmin.dbf' SIZE 25 M AUTOEXTEND ON  
MAXSIZE UNLIMITED;  
create USER strmadmin identified by strmadmin DEFAULT TABLESPACE strmadmin  
QUOTA UNLIMITED ON strmadmin ;
```

3.5 赋权

```
grant dba to strmadmin;  
exec  
DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE('STRMADMIN');
```

3.6 创建 DB link

查看

```
select * from global_name;  
update global_name set global_name='SVEN1'; (在 sys 下的表)  
-- alter database rename global_name to sven1;
```

配置好源端和目标端的 TNS

dblink 名称要和 远端的 global_name 一致, 并且库之间的 global_name 不能相同

源端 :

```
conn strmadmin/ strmadmin;  
create database link SVEN2 connect to strmadmin identified by strmadmin using '10g_2';
```

目标端 :

```
conn strmadmin/ strmadmin;  
create database link SVEN1 connect to strmadmin identified by strmadmin using '10g_1';
```

分别验证:

```
select * from dual@SVEN2;  
select * from dual@SVEN1;
```

3.7 在源端和目标端分别创建队列

源端 :

```
exec  
dbms_streams_adm.set_up_queue(queue_table => 'strmadmin.queue_table', queue_nam  
e => 'strmadmin.queue_table', queue_user => 'STRMADMIN');
```

--创建双向复制需要的应用队列

```
exec
dbms_streams_adm.set_up_queue(queue_table => 'strmadm.apply_table', queue_name => 'strmadm.apply_src', queue_user => 'STRMADMIN');
```

目标端：

创建应用队列：

```
exec
dbms_streams_adm.set_up_queue(queue_table => 'strmadm.queue_table', queue_name => 'strmadm.queue_table', queue_user => 'STRMADMIN');
```

--创建双向复制的捕获队列：

```
exec
dbms_streams_adm.set_up_queue(queue_table => 'strmadm.capture_table', queue_name => 'strmadm.capture_dest', queue_user => 'STRMADMIN');
```

--

删除：

```
exec
DBMS_STREAMS_ADM.REMOVE_QUEUE( queue_name => 'strmadm.10g_1_queue'
);
```

```
exec DBMS_STREAMS_ADM.remove_streams_configuration(); --全部删除
```

3.8 创建进程

源端：

在方案级别指定抓取进程：

```
exec
dbms_streams_adm.add_schema_rules( schema_name => 'hr', streams_type => 'capture'
, streams_name => 'capture_10g_1', queue_name => 'strmadm.queue_table', include_ddl => true, include_ddl => true);
```

--这个其实就是创建规则，and_condition 可以配置创建自定义的过滤。

具体的参数说明可以参考：

http://docs.oracle.com/cd/B28359_01/appdev.111/b28419/d_streams_admin.htm#i997270

--双向复制需要的应用进程：

```
conn strmadmin/strmadmin
```

```
exec
```

```
dbms_streams_admin.add_schema_rules( schema_name => 'hr', streams_type => 'apply', streams_name => 'apply_src', queue_name => 'strmadmin.apply_src', include_dml => true, include_ddl => true, source_database => 'SVEN2', inclusion_rule => true);
```

目标端：

--应用进程：

```
conn strmadmin/strmadmin
```

```
exec
```

```
dbms_streams_admin.add_schema_rules( schema_name => 'hr', streams_type => 'apply', streams_name => 'apply_dest', queue_name => 'strmadmin.queue_table', include_dml => true, include_ddl => true, source_database => 'SVEN1', inclusion_rule => true);
```

--双向复制需要的抓取进程：

```
exec
```

```
dbms_streams_admin.add_schema_rules( schema_name => 'hr', streams_type => 'capture', streams_name => 'capture_dest', queue_name => 'strmadmin.capture_dest', include_dml => true, include_ddl => true);
```

3.9 初始化数据：

```
exp userid=hr/hr@10g_1 file='/tmp/hr.dmp' object_consistent=y rows=y  
imp  userid=system/oracle@10g_2 file='/tmp/hr.dmp' ignore=y commit=y  
log='/tmp/hr.log' streams_instantiation=y fromuser=hr touser=hr
```

使用 expdp/impdp 初始化：

首先 2 边数据库要创建导出导入的文件夹。来进行导出导入。

```
mkdir -p /home/oracle/backup
```

```
CREATE OR REPLACE DIRECTORY my_dir as '/home/oracle/backup';
```

```
GRANT read, write ON DIRECTORY my_dir TO hr ; --或者用要导出的 创建这个文件
```

夹就不用赋权了。

```
expdp hr/hr directory=my_dir dumpfile=hr.dmp nologfile=Y schemas=hr
```

scp 到目标端之后再：

```
impdp hr/hr directory=my_dir dumpfile=hr.dmp nologfile=Y remap_schema=hr:hr
remap_tablespace=users:users table_exists_action=replace
```

也可以使用过程初始化 SCN

```
conn strmadmin/strmadmin
```

```
declare
```

```
v_scn number;
```

```
begin
```

```
v_scn := dbms_flashback.get_system_change_number();
```

```
dbms_apply_adm.set_schema_instantiation_scn@SVEN1(
```

```
source_schema_name => 'hr',
```

```
source_database_name => 'SVEN2',
```

```
instantiation_scn => v_scn,
```

```
recursive => true);
```

```
end;
```

```
/
```

3.10 创建传播进程:

源端：

```
connect strmadmin/strmadmin
```

```
exec
```

```
dbms_streams_adm.add_schema_propagation_rules(schema_name => 'hr', streams_name => 'prop_to_10g_2',
source_queue_name => 'strmadmin.queue_table',
destination_queue_name => 'strmadmin.queue_table@SVEN2', include_dml => true,
include_ddl => true, source_database => 'SVEN1');
```

#修改 propagation 休眠时间为 0，表示实时传播 LCR。

```
exec
```

```
dbms_aqadm.alter_propagation_schedule(queue_name => 'strmadmin.queue_table', destination => 'SVEN2', latency => 0);
```

注意：11g 中如果出现 ORA-24042: no propagation schedule exists for QUEUE

PROD_QUEUE and DESTINATION , 指定 destination_queue =>即可。

目标端：

```
connect strmadmin/strmadmin
```

```
exec
```

```
dbms_streams_adm.add_schema_propagation_rules(schema_name => 'hr',streams_name => 'prop_to_10g_1',source_queue_name => 'strmadmin.capture_dest',destination_queue_name => 'strmadmin.apply_src@SVEN1',include_dml => true,include_ddl => true,source_database=>'SVEN2');
```

#修改 propagation 休眠时间为 0 , 表示实时传播 LCR。

```
exec
```

```
dbms_aqadm.alter_propagation_schedule(queue_name => 'strmadmin.capture_dest',destination => 'SVEN1', latency => 0);
```

--使用 [DISABLE_PROPAGATION_SCHEDULE](http://www.parnassusdata.com) 删除

详细可以参考 dbms_aqadm 的用法

http://docs.oracle.com/cd/B28359_01/appdev.111/b28419/d_aqadm.htm#i1015375

4 stream 启停

4.1 启动

#以 strmadmin 身份，登录从数据库。

```
conn strmadmin/strmadmin
```

#启动 Apply 进程

```
exec dbms_apply_adm.start_apply(apply_name => 'apply_dest');
```

#以 strmadmin 身份，登录主数据库。

```
conn strmadmin/strmadmin
```

#启动 Capture 进程

```
exec dbms_capture_adm.start_capture(capture_name => 'capture_10g_1');
```

---双向复制启动 源端的应用进程，目标端的抓取进程：

源端：

```
exec dbms_apply_adm.start_apply(apply_name => 'apply_src');
```

目标端：

```
exec dbms_capture_adm.start_capture(capture_name => 'capture_dest');
```

4.2 停止

停止捕获进程

```
connect strmadmin/strmadmin
```

```
exec dbms_capture_adm.stop_capture(capture_name => 'capture_10g_1');
```

停止应用进程：

```
connect strmadmin/strmadmin
```

```
exec dbms_apply_adm.stop_apply(apply_name => 'apply_src');
```

--目标端停止：

```
exec dbms_capture_adm.stop_capture(capture_name => 'capture_dest');
```

```
exec dbms_apply_adm.stop_apply(apply_name => 'apply_dest');
```

---在进程启动前要进程初始化，启动顺序最好按照应用-捕获，从源端开始启动。

5 实际测试

1、在源端 hr 用户下创建测试表 test

```
conn hr/hr
```

```
create table test(id number primary key,name varchar2(20));
```

发现在源端和目标端的 alert.log 中出现了

```
Sat Jul 11 23:06:45 2015
```

```
knllgobjinfo: MISSING Streams multi-version data dictionary!!!
```

```
knlldmm: gdbnm=SVEN1
```

```
knlldmm: objn=53313
```

```
knlldmm: objv=1
```

```
knlldmm: scn=722490
```

根据对象 id 发现就是刚创建的对象，在目标端 hr 用户下 查看，test 表 (DDL) 已经复制过去了。这就说明了流数据字典自动重载了。

```
select LVL0NAME,ownername from SYSTEM.LOGMNRC_GTLO where
LVL0NAME='TEST'
```

```
SQL> select LVL0NAME,ownername from SYSTEM.LOGMNRG_GTLO where LVL0NAME='T
EST';

LVL0NAME OWNER
-----
TEST      HR
```

在源端和目标端查询字典对象信息，是可以查到的。这个信息是可以忽略的。

2、源端插入 数据：

```
SQL> conn hr/hr;
```

已连接。

```
SQL> insert into test values(1,'a');
```

已创建 1 行。

```
SQL> commit;
```

目标端也能查询到了：

```
SQL> select * from test;
```

```

      ID NAME
-----
      1 a
```

3、更新删除操作均成功，下面进行双向测试：

在目标端插入数据，更新，删除，DML 均成功复制到源端。

在目标端创建 T2

```
SQL> create table t2 (id number primary key,name varchar2(20));
```

Table created.

源端查询：

```
SQL> desc t2;
```

```

名称                                     是否为空? 类型
-----
ID                                       NOT NULL NUMBER
NAME                                     VARCHAR2(20)
```

目标端删除表 t2, 源端也成功删除。增加字段

4、目标端创建索引：

```
SQL> create index idx_test_1 on test(name);
```

Index created.

源端

```
SQL> SELECT table_name, index_name FROM user_indexes WHERE
table_name='TEST';
```

TABLE_NAME	INDEX_NAME
TEST	IDX_TEST_1
TEST	SYS_C005551

源端删除索引，目标端也被删除。

6.其他信息

6.1 日志信息

强行关闭目标机器，然后再更新源端数据，alter.log 出现

Sun Jul 12 14:57:43 2015

Propagation Schedule for (STRMADMIN.QUEUE_TABLE, SVEN2) encountered following error:

ORA-03135: 连接失去联系

目标端机器启动后，自动恢复，不需要重新配置和启动 stream。源端 DML 应用到目标端。

第一次捕获进程启动的时候再 alert.log 里会发现一些 LOGMNR 索引被自动创建：

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_ATTRCOL\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_CCOL\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_CDEF\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_COL\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_COLTYPE\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_ICOL\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_INDS\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_INDCOMPART\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_INDPART\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_INDSUBPART\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_LOB\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_LOBFRAG\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_OBJ\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_TAB\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_TABCOMPART\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_TABPART\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_TABSUBPART\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_TS\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_TYPES\$ have been marked unusable

Sat Jul 11 23:02:58 2015

Some indexes or index [sub]partitions of table SYSTEM.LOGMNR_USERS\$ have been marked unusable

Indexes of table SYSTEM.LOGMNR_ATTRCOL\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_ATTRIBUTES\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_CCOL\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_CDEF\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_COL\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_COLTYPE\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_DICTIONARY\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_ICOL\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_IND\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_INDCOMPART\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_INDPART\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_INDSUBPART\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_LOB\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_LOBFRAG\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_OBJ\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_TAB\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_TABCOMPART\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_TABPART\$ have been rebuilt and are now usable
Indexes of table SYSTEM.LOGMNR_TABSUBPART\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_TS\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_TYPE\$ have been rebuilt and are now usable

Indexes of table SYSTEM.LOGMNR_USER\$ have been rebuilt and are now usable

Sat Jul 11 23:03:05 2015

LOGMINER: Begin mining logfile for session 21 thread 1 sequence 11,
/home/oracle/arch/1_11_884649572.dbf

Sat Jul 11 23:03:06 2015

LOGMINER: End mining logfile: /home/oracle/arch/1_11_884649572.dbf

Sat Jul 11 23:03:06 2015

LOGMINER: Begin mining logfile for session 21 thread 1 sequence 12,
/u01/app/oracle/ORCL10/redo03.log

Sat Jul 11 23:03:10 2015

LOGMINER: End mining logfile: /u01/app/oracle/ORCL10/redo03.log

Sat Jul 11 23:03:10 2015

LOGMINER: Begin mining logfile for session 21 thread 1 sequence 13,
/u01/app/oracle/ORCL10/redo01.log

Sat Jul 11 23:05:19 2015

之间还出现过下面错误。因为/etc/hosts 没配置 导致监听的 IPC 无法识别

Sat Jul 11 22:50:43 2015

dispatcher 'D000' encountered error getting listening address

Sat Jul 11 22:50:43 2015

Errors in file /u01/app/oracle/admin/ORCL10/bdump/orcl10_ora_5829.trc:

ORA-07445: exception encountered: core dump [kslgetl()+120] [SIGSEGV] [Address not mapped to object] [0x000000208] [] []

ORA-00108: failed to set up dispatcher to accept connection asynchronously

修正 TNS 相关配置，重启数据库后解决。

6.2 参考博客

<http://www.cnblogs.com/rootq/articles/1230415.html>

<http://www.linuxidc.com/Linux/2012-01/51145p3.htm>

5.总结

本次试验主要是对流环境进行搭建，在单项复制的基础上，完成双节点的双向复制。

6.其他问题

日志中出现的控制文件增长的信息，目前还不确定是不是由于大事务导致的控制文件增长的情况，如果了解需要继续查阅资料。

6.1 未解决的问题

问题号	问题描述	解决方案	日期

已解决的问题

问题号	问题描述	解决方案	解决日期

ParnassusData

ParnassusData Corporation, Shanghai, GaoPing Road No. 733. China

Phone: (+86) 400-690-3643

ParnassusData.com

Copyright © 2013, ParnassusData and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, or including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

Copyright © 2015 ParnassusData Corporation. All Rights Reserved.