



目 录

目 录.....	2
第 8 章 GoldenGate 日常维护.....	3
8.1 长事务处理.....	3
8.2 源端和目标端增减复制表.....	5
8.2.1 增加复制表.....	5
8.2.2 修改数据表的结构.....	6
8.2.3 (仅复制 DML 时) 客户应用的升级.....	6
8.2.4 减少复制表.....	7
8.3 数据表重新同步.....	8
8.4 给数据库打补丁.....	9
8.5 给 GoldenGate 程序打补丁.....	9

Beijing Landing Technologies

第 1 章 GoldenGate 日常维护

当 GoldenGate 软件部署实施完毕之后，接下来一部分非常重要的工作就是 GoldenGate 的日常维护与异常处理；这也是非常体现技术含量与一个工程师睿智的环节。

下面我们就来介绍这一部分内容，希望在工作中尽可能的有所帮助。

日常维护包括监视进程的运行状态，调优 GoldenGate 各进程处理性能与排除日常故障。做到这些话我们经常需要利用以下工具获得足够多的诊断信息：

- GGSCI 命令行；
- Oracle GoldenGate 的跟踪参数；
- Oracle GoldenGate 的 report 文件与日志；
- 操作系统的日志；
- 用 logdump 工具分析 trail 文件；

等等。

1.1 长事务处理

1. 检查长交易的存在

在停止抽取进程前需要通过命令检查是否存在长交易，以防止下次启动无法找到归档日志：

```
GGSCI> info extXX, showch (注：查看抽取进程的检查点)
Read Checkpoint #1
Recovery Checkpoint (position of oldest unprocessed transaction in the data
source):
  Thread #: 1
  Sequence #: 9671
  RBA: 239077904
  Timestamp: 2008-05-20 11:39:07.000000
  SCN: 2195.1048654191
  Redo File: Not available

Current Checkpoint (position of last record read in the data source):
  Thread #: 1
  Sequence #: 9671
  RBA: 239377476
  Timestamp: 201-02-20 11:39:10.000000
  SCN: 2195.1048654339
  Redo File: Not Available

Read Checkpoint #2
... ..
```

```
Recovery Checkpoint (position of oldest unprocessed transaction in the data source):
```

```
Thread #: 2
Sequence #: 5287
RBA: 131154160
Timestamp: 2011-02-20 11:37:42.000000
SCN: 2195.1048640151
Redo File: /dev/rredo07
```

```
Current Checkpoint (position of last record read in the data source):
```

```
Thread #: 2
Sequence #: 5287
RBA: 138594492
Timestamp: 2011-02-20 11:39:14.000000
SCN: 2195.1048654739
Redo File: /dev/rredo07
```

示例 8-1

为了方便长交易的管理，GoldenGate 提供了一些命令来查看这些长交易，可以帮助客户和应用开发商查找到对应长交易，并在 GoldenGate 中予以提交或者回滚。

查看长交易的方法：

```
Ggsci> send extract <进程名>, showtrans [thread n] [count n]
```

示例 8-2

其中，<进程名>为所要察看的进程名，如 extsz/extxm/extjx 等；

Thread n 是可选的，表示只查看其中一个节点上的未提交交易；

Count n 也是可选的，表示只显示 n 条记录。

例如，查看 extsa 进程中节点 1 上最长的 10 个交易，可以通过下列命令：

```
Ggsci> send extract extsa, showtrans thread 1 count 10
```

示例 8-3

输出结果是以时间降序排列的所有未提交交易列表，通过 xid 可以查找到对应的事务，请应用开发商和 DBA 帮助可以查找出未提交原因，通过数据库予以提交或者回滚后 GoldenGate 的 checkpoint 会自动向前滚动。

2. 使用 GoldenGate 命令跳过或接受长交易的方法

在 GoldenGate 中强制提交或者回滚指定事务，可以通过以下命令（<>中的为参数）：

```
Ggsci> SEND EXTRACT <进程名>, SKIPTRANS <5.17.27634> THREAD <2>
```

示例 8-4

该命令跳过交易；

```
Ggsci> SEND EXTRACT <进程名>, FORCETRANS <5.17.27634> THREAD <1>
```

示例 8-5

该命令强制认为该交易已经提交。

说明：使用这些命令只会让 GoldenGate 进程跳过或者认为该交易已经提交，但并不改变数据库中的交易，他们依旧存在于数据库中。因此，强烈建议使用数据库中提交或者回滚交易而不是使用 GoldenGate 处理。

3. 配置长交易告警

可以在 Extract 进程中配置长交易告警，参数如下所示：

```
extract extsz
.....
warnlongtrans 12h, checkintervals 10m
```

示例 8-6

以上表示 GoldenGate 会每隔 10 分钟检查一下长交易，如果有超过 12 个小时的长交易，GoldenGate 会在根目录下的 ggserr.log 里面加入一条告警信息。可以通过察看 ggserr.log 或者在 GGSCI 中执行 view ggsevt 命令查看这些告警信息。

以上配置可以有助于及时发现长交易并予以处理。

说明：在 OGG 11g 中，Extract 提供了 BR 参数可以设置每隔一段时间（默认 4 小时）将长交易缓存到本地硬盘（默认 dirtmp 目录下），因此 Extract 只要不停止一般需要的归档日志不超过 8 个小时（极限情况）。但是如果 Extract 停掉后，便无法再自动缓存长交易，需要的归档日志就会依赖于停机时间变长。

1.2 源端和目标端增减复制表

1.2.1 增加复制表

当 GoldenGate 仅打开了 DML 复制时，源端增加复制表的操作步骤为：在 GoldenGate 的进程参数中，如果通过 * 来匹配所有表，因此只要符合 * 所匹配的条件，那么只要在源端建立了表之后 GoldenGate 就能自动复制，无需修改配置文件，但是需要为新增的表添加附加日志。

步骤如下：

```
GGSCI > dblogin userid GoldenGate, password xxx
GGSCI > info trandata <schema>.<table name>
```

示例 8-7

如果不是 enable 则需要手动加入：

```
GGSCI > add trandata <schema>.<table name>
```

示例 8-8

注：（仅对 Oracle 9i）如果该表有主键或者该表不超过 32 列，则显示 enabled 表示添加成功；如果无主键并且列超过 32 列，则可能出现错误显示无法添加则需要手工处理，此时请根据此方法处理来处理：

```
Alter table <table_name> add supplemental log group <group> (column,..)
always;
```

示例 8-9

如果没有使用统配符，则需要在主 Extract、Data Pump 里面最后的 table 列表里加入新的复制表；在目标端 Replicat 的 map 列表同样也加入该表的映射。

然后，新增表请首先在目标端建立表结构。

如果有外键和 trigger，需要在目标表临时禁止该外键和 trigger，并维护在 dirsql 下的禁止和启用这些对象的对应脚本文件。

对于修改了文件的所有源和目标进程，均需重启进程使新的参数生效。

1.2.2 修改数据表的结构

当数据库需要复制的表结构有所改变，如增加列，改变某些列的属性如长度等表结构改变后，可以按照下列步骤执行：

1. 按照本文前面所述操作顺序停止源和目标端各抽取及投递进程（注意停源端抽取要验证一下归档日志是否存在防止无法重起），无需停止 manager 进程；
2. 修改目标表结构；
3. 修改源表结构；
4. 如果表有主键，并且本次修改未修改主键，则可以直接启动源和目标所有进程继续复制，完成本次修改；否则，如果表无主键或者本次修改了主键则需继续执行下列步骤：

```
GGSCI> dblogin userid GoldenGate, password xxxx
GGSCI> delete trandata schema.mytable
GGSCI> add trandata schema.mytable
```

示例 8-10

（仅对 Oracle 9i）如果表超过了 32 列则上述操作可能会报错，此时需要手工进行处理，请参考上一节内容。

5. 重新启动源端和目标端的抓取和复制进程。

1.2.3 （仅复制 DML 时）客户应用的升级

如果是客户的应用进行了升级，导致了源系统表的变化，在不配置 DDL 复制的情况下，需要对 GoldenGate 同步进程进行修改，可以参照以下步骤。

1. 停止源和目标端各抽取及投递进程（注意停源端抽取要验证一下归档日志是否存在防止无法重起），无需停止 manager 进程；
2. 对源系统进行升级；
3. 在目标端将客户升级应用所创立的存储过程、表、function 等操作再重新构建一遍；
4. 在目标端手工禁止建立的 trigger 和外键，并将这些 sql 以及反向维护的（即重新启用 trigger 和外键）SQL 添加到目标端 OGG dirsql 目录下对应的脚本文件里。

注意：在安装实施时，应当将执行的禁止 trigger 和外键的表放到目标 dirsql 下，文件名建议为 disableTrigger.sql 和 disableFK.sql。同时，需要准备一个反向维护（即重新启用 trigger 和外键，建议为 enableTrigger.sql 和 enableFK.sql）SQL，同样放置到目标端 OGG 的 dirsql 目录下，以备将来接管应用时重新启用。

对于升级过程中在源端增加的表，需要为新增的表添加附加日志。

步骤如下：

```
GGSCI > dblogin userid GoldenGate, password *****
GGSCI > info trandata <schema>.<table name>
```

示例 8-11

如果不是 enable 则需要手动加入：

```
GGSCI > add trandata <schema>.<table name>
```

示例 8-12

注：（仅对 Oracle 9i）如果该表有主键或者该表不超过 32 列，则显示 enabled 表示添加成功；若无主键并且列超过 32 列，则可能出现错误显示无法添加则需要手工处理，此时请参照上一小节内容处理。

5. 对于升级过程中可以在源端 drop 掉的表，GoldenGate 缺省复制所有符合通配符条件的表，可以直接在目标端 drop 掉，无需对复制做任何修改；
6. 如果升级过程中修改了主键的表则需继续执行下列步骤；

```
GGSCI> dblogin userid GoldenGate, password xxxx  
GGSCI> delete trandata schema.mytable  
GGSCI> add trandata schema.mytable
```

示例 8-13

（仅对 Oracle 9i）如果表超过了 32 列则上述操作可能会报错，此时需要手工进行处理，请参考上一小节内容处理。

7. 重新启动源端和目标端的抓取和复制进程。

1.2.4 减少复制表

GoldenGate 缺省复制所有符合通配符条件的表，如果有的表不再需要，可以在源端 drop 掉，然后到目标 drop 掉，无需对复制做任何修改。

如果其中几个表依然存在，只是无需 GoldenGate 复制，则可以通过以下步骤排除：

1. 在源端系统上首先验证所需归档日志存在后通过 stop extXX 停止对应的 extXX 进程；
2. 在目标端系统上 GGSCI 中执行 stop repXX 停止目标端的复制进程；
3. 在源端修改 ext 进程的参数文件排除所不复制的表；

在文件定义 table 的行前面加入一行：

```
tableexclude <schema>.<tablename>;
```

示例 8-14

注意写全 schema 和表的名称。

注：如果是没有使用通配符，则直接注释掉该表所在的 table 行即可。

例如：

```
Ggsci> edit param extXX  
.....  
tableexclude ctais2.TMP_*;  
tableexclude ctais2.BAK_*;  
tableexclude ctais2.MLOG$_*;  
tableexclude ctais2.RUPD$_*;  
tableexclude ctais2.KJ_*;  
  
tableexclude myschema.mytable;  
  
table ctais2.*;  
.....
```

示例 8-15

- 在目标端修改 rep 进程参数，同样排除该表：

```
GGSCI>edit param repXX
```

示例 8-16

在 map 前面加入一行：

```
--mapexclude CTAIS2.SHOULIXINXI  
mapexclude myschema.mytable  
MAP ctais2.* ,TARGET ctais2.*;
```

示例 8-17

注：如果是没有使用通配符，则直接注释掉该表所在的 map 行即可。

- 在目标端系统上启动复制进程 repXX

```
GGSCI > start repXX
```

示例 8-18

- 在源端系统上启动源端的抓取进程 extXX

```
GGSCI > start extXX
```

示例 8-19

即可进入正常复制状态。

1.3 数据表重新同步

如果是某些表由于各种原因造成两边数据不一致，需要重新进行同步，可以参照以下步骤：

- 确认需要重新同步的表；
- 验证源端各个抽取进程，重起所需归档日志存在，然后停止抽取进程 ext*以及 dpe*相关进程；
- 停止目标端的 rep 进程；

注意：步骤 4-6 为将源端数据通过 exp/imp 导入到目标端，客户也可以选择其它初始化方式，比如在目标端为源端表建立 dblink，然后通过“create table as select from”的方式初始化目标端表。

- 从生产端数据库中获得当前的 SCN

```
col current_scn for 9999999999999999  
select dbms_flashback.get_system_change_number current_scn from dual;  
CURRENT_SCN  
-----  
1647598704
```

示例 8-20

- 在源端使用 exp 导出该表或者几张表数据。例如：

```
exp ggs/sprite DIRECT=y buffer=6400000 FLASHBACK_SCN=1647598704 \  
CONSTRAINTS=N GRANTS=N TRIGGERS=N \
```

```
file=/GoldenGate/backup/new_tab.dmp log=/GoldenGate/backup/new_tab.log \  
tables=MW_APP.MWT_UD_KW_SJBGJL_JK_20101223
```

示例 8-21

6. 通过 ftp 传输到目标端;
7. 在目标端, 使用 imp 导入数据;

```
nohup imp ggs/sprite file=/GoldenGate/backup/new_tab.dmp  
log=/GoldenGate/backup/new_tab.log fromuser=MW_APP touser=MW_APP commit=y  
ignore=y buffer=52428800 &
```

示例 8-22

8. 如果这些表有外键, 在目标端检查这些外键并禁止它们 (记得维护 dirsql 下的禁止和启用外键的脚本 SQL);
9. 启动源端和目标端的所有 GoldenGate 进程;
10. 使用 Info all 在 GGSCI 中查看各个进程状况。

1.4 给数据库打补丁

这里说的数据库补丁指的是小补丁, 而非数据库大版本的升级。厂商提供给用户的软件补丁的形式多为编译后的库函数。

以 Oracle 为例这里说的补丁主要是指 oneoff patch、PSU 以及 PSU。这些补丁提供的关键的库函数不会变化, 所以在 GoldenGate 这个层面无需进行变更设置。比如数据库的版本的 Oracle 10.2.0.4, 如果需要给数据库应用最新的 PSU, 则 GoldenGate 本身无需变更。但是如果是从 9i 升级到 10g, 则对应的 GoldenGate 也需要进行变更。

1.5 给 GoldenGate 程序打补丁

在 GoldenGate 概念里, 打补丁等同于升级。升级的一般步骤为:

1. 待源端与目标端追平且没有延时, 先停止抽取进程, 接着停止投递, 复制进程;
2. 将新的补丁包程序解压到一个新的目录, 然后覆盖掉原来的 GoldenGate 所在目录的文件;
3. 重新启动 GoldenGate 各进程: start replicat **; start dpe***; start extract ***。