



目 录

目 录.....	2
第 6 章 IBM AIX 平台 Sybase-Oracle 数据库复制.....	3
6.1 目标概述.....	3
6.2 GoldenGate for sybase 在 AIX 5.3 上的安装注意事项.....	3
6.2.1 GoldenGate 在 AIX 操作系统的要求.....	4
6.2.2 GoldenGate 对 Sybase 数据库的要求.....	4
6.3 使用 DEFGEN 生成数据表定义文件.....	5
6.3.1 编辑 defgen 文件.....	5
6.3.2 利用 defgen 工具生成 defgen.prm 文件.....	5
6.3.3 将生成好的数据定义文件 ftp 二进制模式传到容灾端对应的目录 dirdef.....	6
6.4 配置源端进程.....	6
6.4.1 initial data load.....	6
6.4.2 抽取进程与投递进程的配置.....	8
6.4.2.1 配置抽取进程 exta.....	8
6.4.2.2 配置投递进程 dpea.....	8
6.5 配置目标端进程.....	9
6.5.1 在容灾端配置管理进程 MGR.....	9
6.5.2 配置全局参数.....	9
6.5.3 添加检查点表.....	9
6.5.4 编辑复制进程 repa.....	9

Beijing Landing Technologies

第 6 章 IBM AIX 平台 Sybase-Oracle 数据库复制

6.1 目标概述

本节内容为利用实时数据同步软件 GoldenGate 把数据从 Sybase 数据库（源端）实时复制到 Oracle 数据库（目标端）的实施过程，和以往的试验不同的是这次复制源端和目标端的操作系统、数据库平台都不一样。

部分以前的技术可能无法使用，但是总的原理是一致的。注意本章中对于前期准备工作的细节不再详述，部分请参考前面的章节的内容。

复制的原理示意图如下：

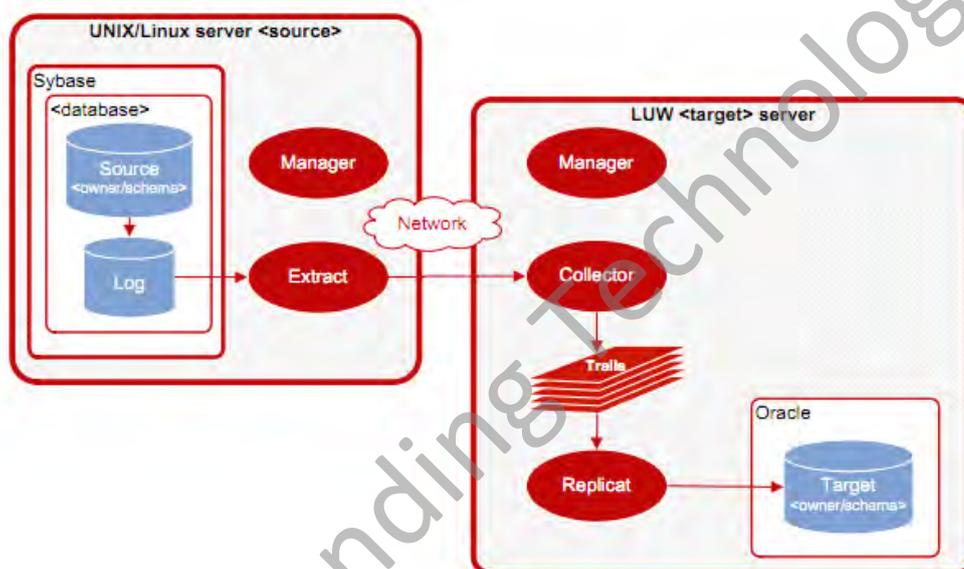


图 6-1

6.2 GoldenGate for sybase 在 AIX 5.3 上的安装注意事项

AIX 5.3 需要 XLC/C++ Runtime v10.1 以上以及 libpthreads version 5.3.0.51 或以上。确认 gg 的环境变量已经设置好，GoldenGate 运行的时候需要用到数据库的一些包，所以需要在 profile 文件里添加相应的 LIBPATH。

步骤概览为：

- a) 源端和目标端准备好环境，包括准备好数据库，安装好 GoldenGate
- b) 将源端历史数据通过 initial data 初始化到容灾端。

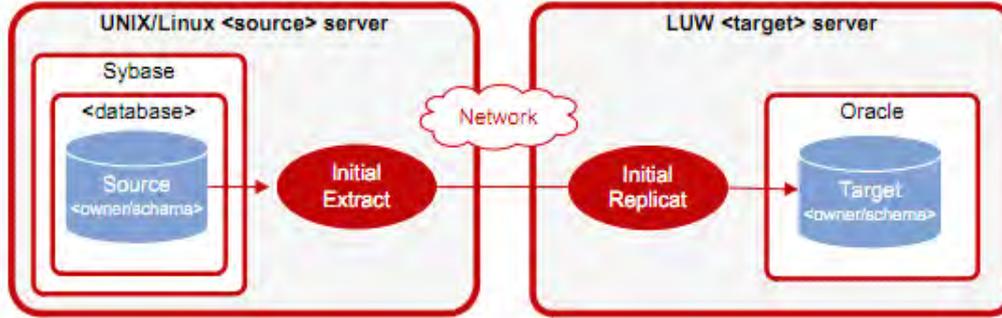


图 6-2

- c) 配置源端的抽取，投递进程。
- d) 配置容灾端的复制进程。

6.2.1 GoldenGate 在 AIX 操作系统的要求

1. 磁盘空间需求：

依据实际的数据大小，确定 GoldenGate 所需要的实际空间大小，一般需要 50-150M 的空间来安装 GoldenGate 软件，然后根据数据的大小确定队列文件的空间，计算公式为：

$[\log \text{ volume in one hour}] \times [\text{number of hours downtime}] \times .4 = \text{trail disk space.}$

2. 确认/etc/hosts 文件 ip 与 hostname 一一对应。
3. 确认 GoldenGate 使用的端口没有被占用，默认为 7840-7850。
4. 确认安装 GoldenGate 的用户对 GoldenGate 有对应的读写权限。

6.2.2 GoldenGate 对 Sybase 数据库的要求

- 1, 需要设置 GoldenGate 将用到的环境变量 DSQUERY
- 2, 抽取进程将用到 sybase 的 replication API
- 3, 抽取进程用 sybase LMT 来读事务日志，所以 RepServer 不可以和 GoldenGate 同运行
- 4, 抽取进程必须允许管理 secondary log truncation point
- 5, source replication server 必须处于激活状态，因为 GoldenGate 不能从热备模式的数据库里捕获数据。

6, 源端 sybase 库需要创建一个数据库用户来管理 GoldenGate，并要赋予相应的权限 sp_role 'grant', replication_role, <Extract user>

此时在源库上简历测试数据，添加表附加日志：

- 1 在 wei 用户下，hello 数据库上，建立测试数据

```
Shell> isql -Uwei -Pshenweigang -SNOE136 < demo_syb_create.sql  
Shell> isql -Uwei -Pshenweigang -SNOE136 < demo_syb_insert.sql
```

示例 6-1

2 设置日志截断点

```
[hello@node136gg]$isql -Uwei -Pshenweigang -SNOE136  
1>use hello  
2>go  
1>dbcc settrunc( 'ltm' , ' valid' )  
2>go
```

示例 6-2

3 添加表级附加日志

```
GGSCI(node136.com29>dblogin sourcedb hello@NOE136,userid wei,password  
shenweigang  
Successfully logged into database.  
GGSCI(node136.com)30>add trandata dbo.*  
Transaction logging is already enables for table "dbo"." TCUS TMER".  
Transaction logging is already enables for table "dbo"." TCUS TORD".  
GGSCI(node136.com)31>info trandata*  
Transaction logging enables for table dbo.TCUS TMER.  
Transaction logging enables for table dbo.TCUS TORD.
```

示例 6-3

6.3 使用 DEFGEN 生成数据表定义文件

利用 DEFGEN 工具可以为源库表或目标端表生成数据定义文件。当源库与目标库类型不一样或者源端的表与目标端的表结构不相同，数据定义文件是必须要有的。

生成数据定义文件的步骤：

6.3.1 编辑 defgen 文件

```
GGSCI> edit param defgen  
DEFDFILE dirdef/source.def, PURGE  
SOURCEDB hello@NOE136, USERID wei, password shenweigang  
TABLE dbo.TCUSTMER;  
TABLE dbo.TCUSTORD;
```

示例 6-4

注意，sourcedb 后边的选项 hello 为数据库名，NOE136 为 server 名，一定要跟实际环境一一对应。

6.3.2 利用 defgen 工具生成 defgen.prm 文件

```
Shell> defgen paramfile dirprm/defgen.prm
```

示例 6-5

生产的数据定义文件内容如下：

```
[hello@node136 ~]$ cd /gg
[hello@node136 gg]$ cd dirdef
[hello@node136 dirdef]$ ls
source.def
[hello@node136 dirdef]$ cat source.def
*
* Definitions created/modified 2011-02-11 15:47
*
* Field descriptions for each column entry:
*
* 1 Name
* 2 Data Type
* 3 External Length
* 4 Fetch Offset
* 5 Scale
* 6 Level
* 7 Null
* 8 Bump if Odd
* 9 Internal Length
* 10 Binary Length
* 11 Table Length
* 12 Most Significant DT
* 13 Least Significant DT
* 14 High Precision
* 15 Low Precision
* 16 Elementary Item
* 17 Occurs
* 18 Key Column
* 19 Sub Data Type
```

图 6-3

6.3.3 将生成好的数据定义文件 ftp 二进制模式传到容灾端对应的目录 dirdef。

```
[oracle@node1 goldengate]$cd dirdef
[oracle@node1 dirdef]$ls
Source.def
```

示例 6-6

6.4 配置源端进程

6.4.1 initial data load

在配置源端进程前，先需要同步初始化数据，这里因为数据库平台不一样，所以不能用传统的同平台数据库相同的方法。一种可行的办法是：把源端数据库导出成文本文件，再导入到目标数据库中。我这里采用另外一种方法即 GoldenGate 自带 initial data load 方式。

在源端执行：

```
GGSCI> ADD EXTRACT E1N1A, SOURCE1STABLE
```

示例 6-7

确认添加成功

```
GGSCI> INFO EXTRACT *, TASKS
```

示例 6-8

编辑 initial load 抽取进程的参数文件

```
GGSCI> EDIT PARAMS EINIA
EXTRACT EINIA
SOURCEDB hello@NODE136, USERID wei, PASSWORD shenweigang
RMTHOST 192.168.0.137, MGRPORT 7839
RMTTASK REPLICAT, GROUP RINIA
TABLE dbo.TCUSTMER;
TABLE dbo.TCUSTORD;
```

示例 6-9

在容灾端也需要配置 rinia 参数文件:

```
GGSCI> EDIT PARAMS RINIA
REPLICAT RINIA
USERID GoldenGate, PASSWORD GoldenGate
DISCARDFILE ./dirrpt/RINIA.dsc, PURGE
SOURCEDEFS ./dirdef/source.def
MAP dbo.TCUSTMER, TARGET SCOTT.TCUSTMER;
MAP dbo.TCUSTORD, TARGET SCOTT.TCUSTORD;
```

示例 6-10

此次需要注意数据定义文件为源端生成的，且远程传过来的文件。

```
GGSCI> ADD REPLICAT RINIA, SPECIALRUN
```

示例 6-11

确认已经添加上

```
GGSCI> INFO REPLICAT *, TASKS
```

示例 6-12

执行初始化进程

```
GGSCI> START EXTRACT EINIA
```

示例 6-13

查看结果:

源端: GGSCI> VIEW REPORT EINIA

```
GGSCI (node136.com) 40> view report einia
.....
*****
**                Running with the following parameters                **
*****
EXTRACT EINIA
SOURCEDB hello@NODE136, USERID wei, PASSWORD *****
```

```

RMTHOST 192.168.0.137, MGRPORT 7839
RMTTASK REPLICAT, GROUP RINIA
TABLE dbo.TCUSTMER;
Using the following key columns for source table dbo.TCUSTMER: CUST_CODE.
TABLE dbo.TCUSTORD;
Using the following key columns for source table dbo.TCUSTORD: CUST_CODE,
ORDER_DATE, PRODUCT_CODE, ORDER_ID.
CACHEMGR virtual memory values (may have been adjusted)
.....
Processing table dbo.TCUSTMER

Processing table dbo.TCUSTORD
*****
*                               ** Run Time Statistics **
*****
Report at 2011-02-11 18:15:55 (activity since 2011-02-11 18:15:50)
Output to RINIA:
From Table dbo.TCUSTMER:
#                inserts:           2
#                updates:           0
#                deletes:           0
#                discards:          0
From Table dbo.TCUSTORD:
#                inserts:           2
#                updates:           0
#                deletes:           0
#                discards:          0
    
```

示例 6-14

此处一定要注意，启动 initial 的时候，只需要在源端启动，容灾端会自动复制，确认完成可以用一下命令：

目标端：GGSCI> VIEW REPORT RINIA

6.4.2 抽取进程与投递进程的配置

6.4.2.1 配置抽取进程 exta

```

GGSCI> ADD EXTRACT exta, TRANLOG, BEGIN NOW
GGSCI> ADD EXTTRAIL /gg/dirdat/ga, EXTRACT exta, MEGABYTES 25
GGSCI> EDIT PARAM exta
extract exta
setenv (DSQUERY = "NODE136")
sourcedb hello@NODE136, USERID wei, PASSWORD weigang
exttrail /gg/dirdat/ga
DYNAMICRESOLUTION
discardfile ./dirrpt/exta.dsc, purge
TABLE dbo.TCUSTMER;
    
```

```
TABLE dbo.TCUSTORD;
```

示例 6-15

这里源库用到的是 hello 这个库，用户名为 wei，传输的表为 demo_syb_create.sql 创建的例子。

查看抽取进程是否添加上

```
GGSCI> info extract exta
```

示例 6-16

6.4.2.2 配置投递进程 dpea

```
GGSCI > edit params dpea
extract dpea
passthru
rmthost 192.168.0.137, mgrport 7809
rmttrail /GoldenGate/dirdat/
discardfile ./dirrpt/dpea.dsc, purge
table dbo.*;
```

示例 6-17

这里的容灾端用的是局域网的另一台机器。

```
[oracle@node1 dirdef]$ hostname
node1
```

图 6-4

6.5 配置目标端进程

6.5.1 在容灾端配置管理进程 MGR

```
PORT 7809
DYNAMICPORTLIST 7840-7849
AUTOSTART EXTRACT *
AURESTART EXTRACT *, RETRIES 5, WAITMINUTES 3
PURGEOLDEXTRACTS ./dirdat/*, usecheckpoints, minkeepdays 3
LAGREPORHOURS 1
LAGINFOMINUTES 30
LAGCRITICALMINUTES 45
```

示例 6-18

6.5.2 配置全局参数

```
Shell> GGSCI
GGSCI> EDIT PARAMS ./GLOBALS
CHECKPOINTTABLE GoldenGate.ggschkpt
GGSCI> EXIT
```

示例 6-19

6.5.3 添加检查点表

```
GGSCI (node1) 1> dblogin userid GoldenGate, password GoldenGate
Successfully logged into database.
GGSCI (node1) 2> ADD CHECKPOINTTABLE
```

示例 6-20

6.5.4 编辑复制进程 repa

```
GGSCI (node1) 12> edit param repa
REPLICAT repa
USERID GoldenGate, PASSWORD GoldenGate
SOURCEDEFS ./dirdef/source.def
DISCARDFILE ./dirrpt/eora.DSC, PURGE
MAP dbo.TCUSTMER, TARGET scott.tcustmer;
MAP dbo.TCUSTORD, TARGET scott.tcustord;
```

示例 6-21

```
GGSCI (node1) 12> info all
```

Program	Status	Group	Lag	Time Since Chkpt
MANAGER	RUNNING			
REPLICAT	RUNNING	REPA	00:00:00	00:00:01

图 6-5