



目 录

目 录.....	2
第 7 章 实际应用中常见场景及案例分析.....	3
7.1 目标概述.....	3
7.2 一对多复制.....	3
7.2.1 单源到多目标复制.....	4
7.2.1.1 独立 Extract 模式 :	4
7.2.1.2 共享 Extract 模式.....	6
7.2.2 单表到多表复制.....	8
7.3 多对一复制.....	8
7.3.1 多源到单目标复制.....	10
7.3.2 多表到单表复制.....	12
7.4 级联复制.....	12
7.4.1 生产库配置.....	13
7.4.2 中间库配置.....	14
7.4.3 目标库配置.....	15
7.5 数据的转换.....	15
7.5.1 数据选择与过滤.....	16
7.5.2 列映射.....	17
7.5.2.1 table-level 列映射.....	17
7.5.2.2 全局列映射.....	19
7.5.2.3 映射数据类型.....	20
7.5.3 函数功能.....	20
7.5.3.1 函数的使用案例.....	20
7.5.3.2 使用@COMPUTE 函数.....	21
7.5.3.3 使用@COLSTAT 函数.....	21
7.5.3.4 使用@COLTEST 函数.....	21
7.5.3.5 使用@IF 函数.....	22
7.5.3.6 使用@CASE 函数.....	22
7.5.3.7 使用@EVAL 函数.....	22
7.6 双业务中心场景.....	22
7.6.1 Primary-Standby 模式切换.....	23
7.6.1.1 配置 Primary-Standby 需要考虑的问题.....	23
7.6.1.2 有计划迁移应用.....	24
7.6.1.3 数据库意外宕机切换.....	27
7.6.2 配置双向复制 Active-Active 模式.....	30
7.6.3 Active-active 冲突处理及解决.....	33

第 7 章 实际应用中常见场景及案例分析

7.1 目标概述

本章介绍 GoldenGate 常见的场景及一些案例的配置方法。

由于在生产环境中很多企业对于生产数据会有各种各样的需要：数据仓库，报表系统，容灾等。如果把所有的应用都在一个生产库上实现，势必会给生产库带来很多的压力。为了满足客户多样的需求，GoldenGate 有与之对应的灵活的解决方案。例如可以对数据进行过滤转换，也可以把多个源库的数据汇总到一个数据库。

GoldenGate 的配置相当的灵活：

灵活的拓扑结构

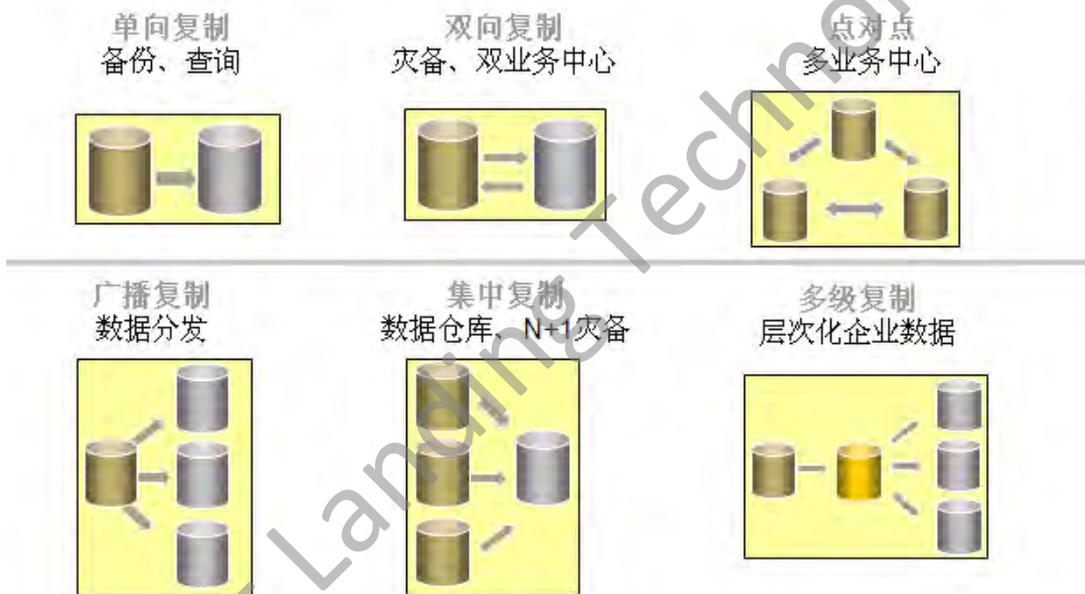


图 7-1

7.2 一对多复制

GoldenGate 支持把一个数据库中的数据同时复制到多个数据库中。这种方式可以让每个系统都得到同样的数据(多地多分数据容容灾)，也可以使用过滤操作让每个系统只复制用户希望的数据(报表系统，只把用户需要的数据复制过去)。

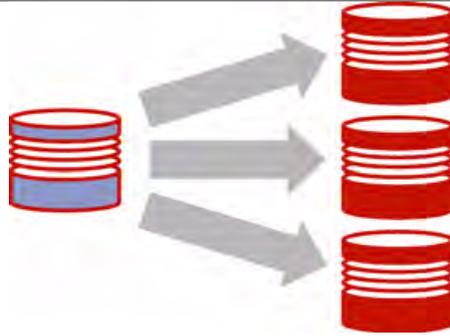


图 7-2

7.2.1 单源到多目标复制

配置单源到多目标可以有两种方法：独立 Extract 模式和共享 Extract 模式。

7.2.1.1 独立 Extract 模式：

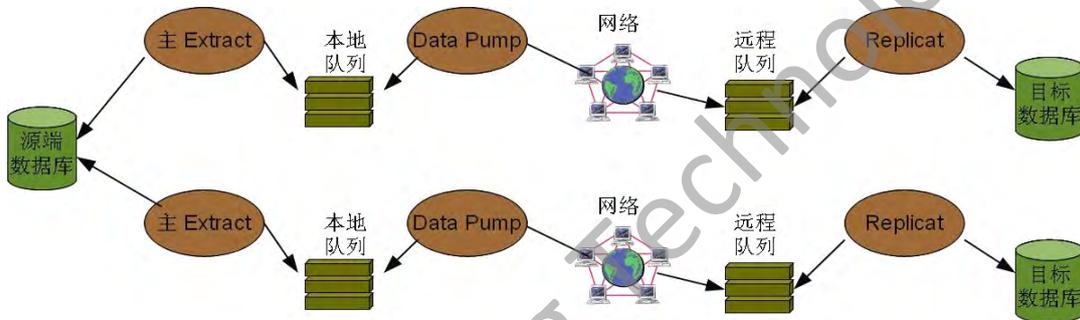


图 7-3

源端针对每个目标库配置独立的 Extract

优势是各复制通道无耦合，互相独立，一条链路抽取链路的中断不会影响其他链路的复制，可独立管理以及数据库产生日志非常大的情况

劣势是需要多个队列，如数据集有重复会占用较多磁盘空间，同时多个 Extract 对于主机资源占用较多

适用于各目标所需数据集无重合或者重合较小并且主机资源充裕的场景

这种方式是几条完整的 Extract, Pump, Replicat 链路的结合，互不影响，我这里介绍一条完整链路的配置方法，其他链路的配置完全一样。下面是一组 Extract, Pump, Replicat 典型的参数配置，其他方式只要略做修改，就可以使用。

生产端和容灾端创建 MGR 进程：

```
GGSCI (OE5) 1>edit params mgr

PORT 7839
DYNAMICPORTLIST 7840-7849
--AUTOSTART EXTRACT *
--AUTORESTART EXTRACT *,RETRIES 5,WAITMINUTES 3
PURGEOLDEXTRACTS ./dirdat/*,usecheckpoints, minkeepdays 100
```

```
LAGREPORHOURS 1
LAGINFOMINUTES 30
LAGCRITICALMINUTES 45
```

示例 7-1

生产端配置 Extract 进程参数:

```
GGSCI (OE5) 2> edit params extma

EXTRACT extma
userid GoldenGate, password GoldenGate
setenv (NLS_LANG="AMERICAN_AMERICA.ZHS16GBK")
GETTRUNCATES
REPORTCOUNT EVERY 1 MINUTES, RATE
numfiles 50000
DISCARDFILE ./dirrpt/extma.dsc, APPEND, MEGABYTES 50
WARNLONGTRANS 2h, CHECKINTERVAL 3m
EXTTRAIL ./dirdat/ma
DBOPTIONS ALLOWUNUSEDCOLUMN
TRANLOGOPTIONS CONVERTUCS2CLOBS
DYNAMICRESOLUTION
TABLE scott.* ;
SEQUENCE scott.*;
```

示例 7-2

生产端添加 Extract 进程组:

```
GGSCI (OE5) 3>add extract extma, tranlog, begin now
GGSCI (OE5) 4>add exttrail ./dirdat/ma, extract extma, megabytes 500
```

示例 7-3

生产端配置 Pump 进程参数:

```
GGSCI (OE5) 5>edit params dpema

EXTRACT dpema
RMTHOST ip_addr , MGRPORT 7839, compress
PASSTHRU
numfiles 50000
RMTRAIL ./dirdat/ma
DYNAMICRESOLUTION
TABLE scott.* ;
SEQUENCE scott.* ;
```

示例 7-4

生产端添加 Pump 进程组:

```
GGSCI (OE5) 6> ADD EXTRACT dpema ,EXTTRAILSOURCE ./dirdat/ma
GGSCI (OE5) 7> ADD RMTRAIL ./dirdat/ma, EXTRACT dpema, MEGABYTES 500
```

示例 7-5

容灾端配置 Replicat 进程参数：

```
GGSCI (OE5) 2>edit params repma
REPLICAT repma
USERID GoldenGate, PASSWORD GoldenGate
setenv (NLS_LANG="AMERICAN_AMERICA.ZHS16GBK")
--REPORT AT 01:59
REPORTCOUNT EVERY 30 MINUTES, RATE
REPERROR DEFAULT, abend
numfiles 50000
DBOPTIONS ALLOWUNUSEDCOLUMN
MAXTRANSOPS 500000
GROUPTRANSOPS 10000
CHECKPOINTSECS 40
--HANDLECOLLISIONS
assumtargetdefs
DISCARDFILE ./dirrpt/repma.dsc, APPEND, MEGABYTES 1000
GETTRUNCATES
ALLOWNOOPUPDATES
MAP scott.* , TARGET scott.* ;
```

示例 7-6

容灾端添加 checkpoint 表：

```
GGSCI (OE5) 5>edit params ./GLOBALS
Edit params ./GLOBALS
checkpointtable GoldenGate.ckpt
dblogin userid GoldenGate, password GoldenGate
add checkpointtable GoldenGate.ckpt
```

示例 7-7

容灾端添加 Replicat 进程组：

```
GGSCI (OE5) 6> ADD REPLICAT repma ,EXTTRAIL ./dirdat/ma checkpointtable
GoldenGate.ckpt
```

示例 7-8

7.2.1.2 共享 Extract 模式

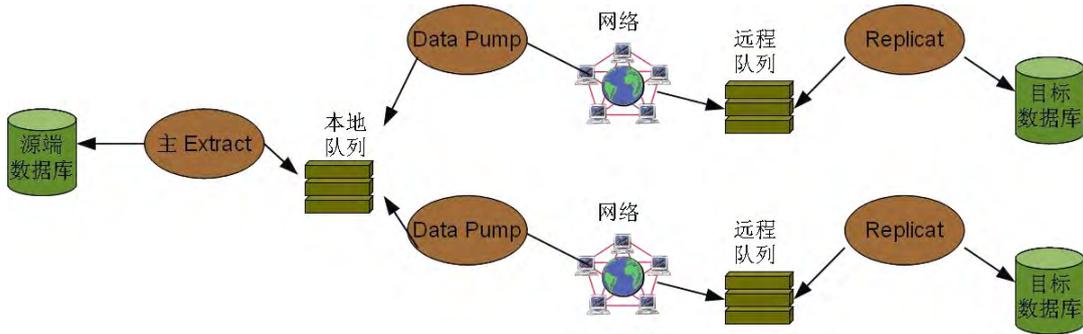


图 7-4

源端对多个目标配置统一的 Extract，然后通过不同 Data Pump 分发数据。

优势是只需一个本地队列，占用空间少，且单个 Extract 占用主机资源少。

劣势是 Extract 作为共享组件出现问题后所有复制均中断。

适用于多个目标所需数据集相同或有较大重合的场景

对于单源到多目标的复制：主要通常 DATA PUMP 进行数据分发。

共享 Extract 的配置方法：

- 1、 生产端和容灾端配置 MGR 进程参数
- 2、 在生产端添加 Extract 进程组和本地 trail 文件

```
ADD EXTRACT <ext>, TRANLOG, BEGIN <time>, [, THREADS]
ADD EXTTRAIL <local_trail>, EXTRACT <ext>
```

示例 7-9

- 3、 编辑 Extract 进程组参数文件，这是只是最简单的参数，可以根据上个案例详细配置

```
EXTRACT <ext>
[SOURCEDB <dsn_1>][USERID <user>[, PASSWORD <pw>]]
EXTTRAIL <local_trail>
TABLE <owner>.<table>;
```

示例 7-10

- 4、 添加两个 Pump 进程组，这是这个案例的关键，通过两个 Pump 进程进行数据转发

```
ADD EXTRACT <pump_1>, EXTTRAILSOURCE <local_trail>, BEGIN <time>
ADD EXTRACT <pump_2>, EXTTRAILSOURCE <local_trail>, BEGIN <time>
```

示例 7-11

- 5、 添加两个远端 trail 队列

```
ADD RMTTRAIL <remote_trail_1>, EXTRACT <pump_1>
ADD RMTTRAIL <remote_trail_2>, EXTRACT <pump_2>
```

示例 7-12

- 6、 配置两个 Pump 进程组的参数文件

```

-----Pump 1-----
EXTRACT <ump_1>
[SOURCEDB <dsn_1>][USERID <user>[, PASSWORD <pw>]]
RMTHOST <target_1>, MGRPORT <portnumber>
RMTTRAIL <remote_trail_1>
[PASSTHRU | NOPASSTHRU]
TABLE <owner>.<table>;
-----pump 2-----
EXTRACT <ump_2>
[SOURCEDB <dsn_1>][USERID <user>[, PASSWORD <pw>]]
RMTHOST <target_2>, MGRPORT <portnumber>
RMTTRAIL <remote_trail_2>
[PASSTHRU | NOPASSTHRU]
TABLE <owner>.<table>;
    
```

示例 7-13

7、在容灾端添加两个 Replicat 进程

```

ADD REPLICAT <rep_1>, EXTTRAIL <remote_trail_1>, BEGIN <time>
ADD REPLICAT <rep_2>, EXTTRAIL <remote_trail_2>, BEGIN <time>
    
```

示例 7-14

8、配置容灾端每个 Replicat 的进程参数

```

-----Replicat 1-----
REPLICAT <rep_1>
SOURCEDEFS <full_pathname> | ASSUMETARGETDEFS
[TARGETDB <dsn_2>][USERID <user id>[, PASSWORD <pw>]]
REPERROR (<error>, <response>)
MAP <owner>.<table>, TARGET <owner>.<table>[, DEF <template name>];
-----Replicat 2-----
REPLICAT <rep_2>
SOURCEDEFS <full_pathname> | ASSUMETARGETDEFS
[TARGETDB <dsn_3>][USERID <user id>[, PASSWORD <pw>]]
REPERROR (<error>, <response>)
MAP <owner>.<table>, TARGET <owner>.<table>[, DEF <template name>];
    
```

示例 7-15

7.2.2 单表到多表复制

单表到多表的复制和单源到多目标的复制非常的相似。其中单源到多目标数据库的复制，看其内部就是许多单表到多目标的复制。单表到多表的复制在任何模式的数据库复制中都可以配置。只需要在容灾端配置 MAP 参数即可。

单表到多表的模式：

- 单表到相同（不同）数据库相同 SCHEMA 下不同表名的复制；
- 单表到相同（不同）数据库不同 SCHEMA 下相同表名的复制；

- 单表到相同（不同）数据库不同 SCHEMA 下不同表名的复制。

下面的参数配置假定把一张表复制到了三个表中的配置方法。

```
MAP lifeman.gg_pol_ben , TARGET ELISDATA.pol_ben;  
MAP lifeman.gg_pol_ben , TARGET ELISDATA.POL_INSURED , colmap ( usedefaults,  
CLIENTNO = INSNO);  
MAP lifeman.gg_pol_ben , TARGET ELISDATA.POL_JNT_INSURED, colmap  
( usedefaults, CLIENTNO = JNT_INSNO);
```

示例 7-16

本例中所有目标表主键与源表主键相同，如不同则需使用 keycols 指定。

单表拆分到单表（多表）的复制：

```
Replicat 1:  
MAP sales.acct, TARGET sales.acct, FILTER (@RANGE (1, 2));  
Replicat 2:  
MAP sales.acct, TARGET sales.acct, FILTER (@RANGE (2, 2));
```

示例 7-17

Beijing Landing Technologies