**exBase**

**启用增量前置步骤文档**

**(Version 3.11)**

**改版记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本** | **发布日期** | **描述** | **作者** | **复审** | **批准** |
| v1.0 | 28/10/2022 | 增量前置步骤说明 | 杨文丽 | 林慧莹 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **变更编号** | **日期** | **变更项** | **描述** | **基线版本** | **变更请求编号** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

【版权声明】

©2007-2022 北京海量数据技术股份有限公司 版权所有

本文档著作权归 **北京海量数据技术股份有限公司**（简称“海量数据”）所有，未经海量数据事先书面许可，任何主体不得以任何形式复制、 修改、抄袭、传播全部或部分本文档内容。

北京海量数据技术股份有限公司保留所有的权利。

【服务声明】

本文档意在向客户介绍海量数据全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的产品、服务的种类、服务标准等应由您与海量数据之间的商业合同约定，除非双方另有约定，否则，海量数据对本文档内容不做任何明示或模式的承诺或保证。

目录

[1. 正向增量迁移前置步骤 1](#_Toc22079)

[1.1. MySQL到目标库正向（通用） 1](#_Toc22889)

[1.1.1. 前提条件 1](#_Toc25675)

[1.1.2. 操作步骤 1](#_Toc6384)

[1.2. Oracle到目标库正向（通用） 3](#_Toc980)

[1.2.1. 前提条件 4](#_Toc20133)

[1.2.2. 操作步骤 4](#_Toc6633)

[1.3. SQL Server到Vastbase G100正向 9](#_Toc2665)

[1.3.1. 前提条件 9](#_Toc435)

[1.3.2. 操作步骤 10](#_Toc16994)

[1.4. Vastbase G100到MySQL正向 13](#_Toc18909)

[1.4.1. 前提条件 13](#_Toc24389)

[1.4.2. 操作步骤 14](#_Toc16945)

[1.5. Vastbase G100到Vastbase G100正向 16](#_Toc31735)

[1.5.1. 前提条件 16](#_Toc824)

[1.5.2. 创建相关表 17](#_Toc31529)

[1.6. Vastbase G100到Kafka正向 18](#_Toc10217)

[1.6.1. 前提条件 18](#_Toc27355)

[1.6.2. 操作步骤 18](#_Toc29090)

[1.6.3. 注意事项 23](#_Toc11496)

[1.6.4. FAQ 23](#_Toc4345)

[2. 反向增量迁移前置步骤 24](#_Toc21771)

[2.1. MySQL到Vastbase G100反向 24](#_Toc25142)

[2.1.1. 前提条件 24](#_Toc28673)

[2.1.2. 操作步骤 25](#_Toc8814)

[2.2. MySQL到Vastbase E100反向 26](#_Toc17580)

[2.2.1. 前提条件 26](#_Toc2707)

[2.2.2. 操作步骤 27](#_Toc26473)

[2.3. MySQL到PostgreSQL反向 29](#_Toc26620)

[2.3.1. 前提条件 29](#_Toc30962)

[2.3.2. 操作步骤 30](#_Toc17967)

[2.4. Oracle到Vastbase G100反向 33](#_Toc30090)

[2.4.1. 前提条件 33](#_Toc7155)

[2.4.2. 操作步骤 33](#_Toc24128)

[2.5. Oracle到Vastbase E100反向 35](#_Toc15855)

[2.5.1. 前提条件 35](#_Toc5515)

[2.5.2. 操作步骤 35](#_Toc11118)

[2.6. Oracle到PostgreSQL反向 38](#_Toc20881)

[2.6.1. 前提条件 38](#_Toc11765)

[2.6.2. 操作步骤 38](#_Toc22994)

[2.7. SQL Server到Vastbase G100反向 41](#_Toc19149)

[2.7.1. 前提条件 41](#_Toc13078)

[2.7.2. 操作步骤 41](#_Toc15802)

# 正向增量迁移前置步骤

## MySQL到目标库正向（通用）

本小节介绍的是以MySQL为源库开启正向增量迁移功能的前置步骤，因为当目标库为Vastbase G100、Vastbase E100、PostgreSQL、GaussDB(for openGauss) 时前置操作一致，因此本小节内容通用。

### 前提条件

* MySQL正向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| MySQL 5.5 | Vastbase G100 V2.2 Build 5、V2.2 Build 10 |
| MySQL 5.6、5.7 、8.0 | Vastbase G100 V2.2 Build 5、V2.2 Build 10 |
| Vastbase E100 V3.0 |
| PostgreSQL 11g、12c |
| MySQL 5.7、8.0 | GaussDB(for openGauss)  V500R002C10 |

### 操作步骤

#### MySQL用户授权

MySQL增量功能需要在源库进行授权和操作，涉及与配置数据源相关信息如下图所示，根据实际填写情况调整。



在exBase数据源连接管理中使用的MySQL用户需要有相应的权限。授权语句：

GRANT SELECT,REPLICATION SLAVE,REPLICATION CLIENT ON \*.\* TO 'MySQL用户'@'%';

GRANT SELECT,INSERT,UPDATE ON MySQL用户.\* TO 'MySQL用户'@'%';

GRANT PROCESS ON \*.\* to 'MySQL用户'@'%';

FLUSH PRIVILEGES;

**说明：**授权语句中的<MySQL用户>指的是连接MySQL数据源时的用户名，根据实际情况进行修改。

#### MySQL配置

* 在MySQL的配置文件my.cnf中配置参数，完成后重启数据库。

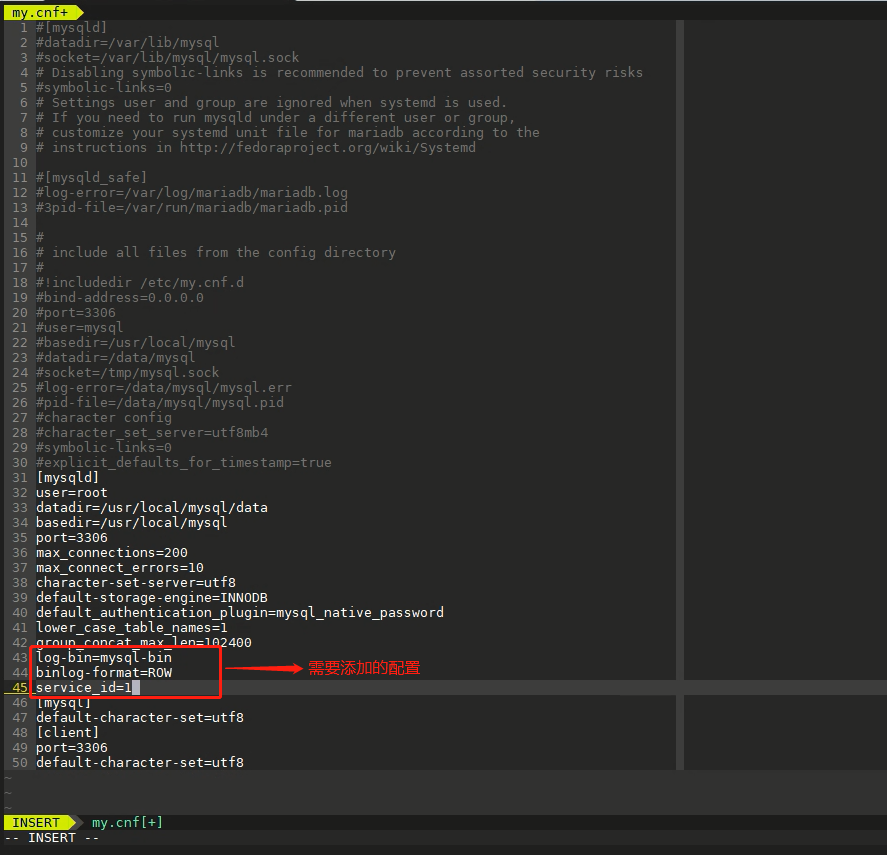
log-bin=MySQL-bin #开启 binlog

binlog-format=ROW #binlog选择ROW模式

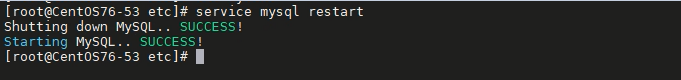
server\_id=1 # 配置服务器id，不必一定是1，集群里的机器不要跟其它机器的相同。

* 示例操作步骤：

**步骤1** 编辑/etc/my.cnf配置文件并添加内容：



**步骤2** 保存后使用命令service MySQL restart重启MySQL数据库：



#### 创建相关表

MySQL正向增量功能的前置步骤需要在源库MySQL及目标库Vastbase G100、Vastbase E100、PostgreSQL、GaussDB(for openGauss)  创建增量相关表。

**步骤1** 登录MySQL数据库，登录用户需要与后续exBase平台使用的MySQL数据源连接用户保持一致。

**步骤2** 新建名为MySQL数据源连接用户同名的数据库。

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE DATABASE ${username};

**步骤3** 切换至MySQL数据源连接用户同名的database下执行以下sql脚本。

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE TABLE ${username}.`incremental\_offset` (

`topic` varchar(128) DEFAULT NULL,

`collectoffset` bigint(20) DEFAULT NULL,

`sourcetime` datetime DEFAULT NULL,

`startlpn` varchar(128) DEFAULT NULL,

`collectcommitlpn` varchar(128) DEFAULT NULL,

`masterid` varchar(128) DEFAULT NULL,

`startlogfile` varchar(128) DEFAULT NULL,

`startlogposition` bigint(20) DEFAULT NULL,

`collectcommitlogfile` varchar(128) DEFAULT NULL,

`collectcommitlogposition` bigint(20) DEFAULT NULL,

`startgtidset` varchar(2048) DEFAULT NULL,

`collectcommitgtidset` varchar(2048) DEFAULT NULL,

`jobid` varchar(128) DEFAULT NULL,

`updatetime` timestamp NULL DEFAULT NULL,

`collecttime` timestamp NULL DEFAULT NULL,

`fullmigratetime` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE ${username}.`tb\_MySQL\_tableddl` (

`contextid` bigint(20) NOT NULL,

`schemaname` varchar(128) DEFAULT NULL,

`tablename` varchar(128) DEFAULT NULL,

`ddl` text

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE ${username}.`tb\_MySQL\_logcontext` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`jobid` varchar(128) DEFAULT NULL,

`masterid` varchar(128) DEFAULT NULL,

`logfile` varchar(128) DEFAULT NULL,

`logposition` varchar(128) DEFAULT NULL,

`updatetime` timestamp NULL DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**步骤4** 登录目标库，登录用户需要与后续exBase平台使用目标库的数据源连接用户保持一致。切换至目标库连接数据源的database、数据源连接用户的schema下执行以下sql脚本：

drop table if exists public.kafkaoffset;

create table public.kafkaoffset (

jobid varchar (128) primary key,

topic varchar (64),

lastoffset bigint,

lastsuboffset bigint,

last\_scn\_number bigint,

scnnumber bigint,

transaction\_id text,

applytime timestamp,

updatetime timestamp,

commitposition text

);

**步骤5** 目标库需要将public.kafkaoffset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE ON public.kafkaoffset TO 目标库用户名;

## Oracle到目标库正向（通用）

本小节介绍的是以Oracle为源库开启正向增量迁移功能的前置步骤，因为当目标库为Vastbase G100、Vastbase E100、PostgreSQL、MySQL、GaussDB(for openGauss) 时前置操作一致，因此本小节内容通用。

### 前提条件

* Oracle正向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| Oracle 11g（11.2.0.4以上）  Oracle 12c（支持CDB）  Oracle 18c  Oracle 19c | Vastbase G100 V2.2 Build 5、V2.2 Build 10 |
| Vastbase E100 V3.0 |
| PostgreSQL 11g、12c |
| GaussDB(for openGauss)  V500R002C10 |
| MySQL 5.6、5.7、8.0 |
| openGauss集中式（企业版）3.0 |
| Oracle 12c（支持PDB） | Vastbase G100 V2.2 Build 5、V2.2 Build 10 |

### 操作步骤

#### Oracle用户授权

* Oracle增量功能需要在源库进行授权和操作，涉及与配置数据源相关信息下图所示，根据实际填写情况调整。



* Oracle在 exBase 的连接用户需要有相应的权限。

|  |  |
| --- | --- |
| **权限名称** | **用途** |
| connect | 系统用户可连接 |
| select any table | 系统需要SELECT所有表,系统元数据发现需要用此权限 |
| select any dictionary | 用于查看任何数据字典,系统需要查询生产数据源的版本，字符集，表，字段等信息，系统元数据发现需要用此权限 |
| select\_catalog\_role | 用于查看一些数据字典的视图，查看role定义操作需要用到此权限 |
| Execute on dbms\_metadata | 获取表结构需要该权限 |
| select any sequence | 获取序列需要该权限 |
| alter any trigger | 获取触发器需要该权限 |

#### Oracle配置

* **连接Oracle数据库**

在命令行工具中执行以下命令以sys用户连接到数据库。

sqlplus /nolog

CONNECT sys/password@host:port AS SYSDBA;

其中：

* password为数据库sys用户的密码，可向数据库管理员获取。
* host为数据库实例所在服务器的IP地址，请根据实际情况设置。
* port为数据库实例所使用的端口，请根据实际情况设置。

**说明：**在实际使用过程中，可以有多种方式连接数据库，此处以命令行方式为例进行说明。

* **开启日志归档和XStream**

**步骤1** 执行以下命令开启Xstream。

alter system set enable\_goldengate\_replication=true;

**步骤2** 检查日志归档是否已开启。

archive log list;

* 若回显打印“Database log mode: No Archive Mode”，说明日志归档未开启，继续执行下一步。
* 若回显打印“Database log mode: Archive Mode”，说明日志归档已开启，直接跳到步骤6。

**步骤3** 执行以下命令配置归档日志参数。

alter system set db\_recovery\_file\_dest\_size = 100G;

alter system set db\_recovery\_file\_dest = '/opt/oracle/oradata/recovery\_area' scope=spfile;

其中：

* 100G为日志文件存储空间的大小，请根据实际情况设置。
* /opt/oracle/oradata/recovery\_area为日志存储路径，请根据实际规划设置，但须确保路径提前创建并有对应oracle操作系统用户读写权限。

**步骤4** 开启日志归档。

**注意：**

* 开启日志归档功能需重启数据库，重启期间将导致业务中断，请谨慎操作。
* 归档日志会占用较多的磁盘空间，若磁盘空间满了会影响业务，请定期清理过期归档日志。

shutdown immediate;

startup mount;

alter database archivelog;

alter database open;

**步骤5** 确认日志归档是否已成功开启。

archive log list;

当回显打印“Database log mode: Archive Mode”，说明日志归档已开启。

**步骤6** 执行以下命令退出数据库连接。

exit;

* **创建XStream用户并赋予权限**

**步骤1** 在命令行工具中执行以下命令，以sys用户连接到数据库实例。

sqlplus sys/password@host:port/SID as sysdba

其中：

* password为数据库sys用户的密码，请向数据库管理员获取。
* host为数据库实例所在服务器的IP地址，请根据实际情况设置。
* port为数据库实例所使用的端口，请根据实际情况设置。
* SID为要同步数据所在实例的实例名，请根据实际情况设置。

**步骤2** 执行以下命令创建 XStream 管理员用户。

说明：该管理员用户即为 exBase 连接 Oracle 数据源所需用户若已存在该用户，可以跳过创建步骤，直接执行步骤3授权。

CREATE TABLESPACE xstream\_adm\_tbs DATAFILE '/opt/oracle/oradata/orcl/xstream\_adm\_tbs.dbf' SIZE 25M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE USER xstrmadmin IDENTIFIED BY password DEFAULT TABLESPACE xstream\_adm\_tbs QUOTA UNLIMITED ON xstream\_adm\_tbs;

其中：

* xstream\_adm\_tbs为XStream管理员用户的表空间名，请根据实际规划设置。
* /opt/oracle/oradata/orcl/xstream\_adm\_tbs.dbf为XStream管理员用户的表空间文件，请根据实际规划设置。
* xstrmadmin为XStream管理员用户名，对应exBase配置连接数据源的用户名,请根据实际规划设置。
* password为XStream管理员用户密码，对应exBase配置连接数据源的用户密码,请根据实际规划设置。

**步骤3** 执行以下命令配置 XStream 管理员所需权限

GRANT CREATE SESSION TO xstrmadmin;

BEGIN

DBMS\_XSTREAM\_AUTH.GRANT\_ADMIN\_PRIVILEGE(

grantee =>'xstrmadmin',

privilege\_type => 'CAPTURE',

grant\_select\_privileges => TRUE,

container=> 'ALL'

);

END;

/

**说明：**“container => 'ALL'”仅当Oracle为12c及以上版本时，才需要添加，否则删除此行内容。

GRANT CREATE SESSION TO xstrmadmin;

GRANT SELECT ON V\_$DATABASE to xstrmadmin;

GRANT FLASHBACK ANY TABLE TO xstrmadmin;

GRANT SELECT ANY TABLE to xstrmadmin;

GRANT LOCK ANY TABLE TO xstrmadmin;

grant select\_catalog\_role to xstrmadmin;

**步骤4** 执行以下命令修改日志记录参数。

alter database add supplemental log data (primary key) columns;

**步骤5** 执行以下命令退出数据库连接。

exit;

* **配置Oracle Streams Pool（可选）**

Oracle Streams Pool是Oracle Streams 使用的System Global Area(SGA) 的一部分内存。此部分内存用于capture，apply，XStream outbound server，也用于缓存缓冲队列的信息。

可以通过以下几种方式决定Oracle Streams pool的大小：

* 使用Automatic Memory Management设置Oracle Streams pool

**步骤1** AMM的启用

在安装过程中，指定Oracle使用内存的百分比的取值就作为MEMORY\_TARGET和MEMORY\_MAX\_TARGET的初始取值使用。

当这两个参数设置为非零取值，那么Oracle就是采用AMM管理策略的。当设置这两个参数为0，则AMM自动关闭。对应的SGA\_TARGET、PGA\_AGGREGATE\_TARGET参数取值非零之后，Oracle自动退化使用ASMM特性。

其中：

* MEMORY\_MAX\_TARGET：定义MEMORY\_TARGET的上限
* MEMORY\_TARGET：定义SGA和PGA总和的上限

**步骤2** 使用sysdba用户登录：

sqlplus / as sysdba

**步骤3** 当MEMORY\_TARGET或MEMORY\_MAX\_TARGET初始化参数设置为非零值时，Automatic Memory Management将自动管理Oracle Streams pool的大小。设置MEMORY\_MAX\_TARGET和MEMORY\_TARGET参数值：

alter system set memory\_max\_target=2G scope = spfile;

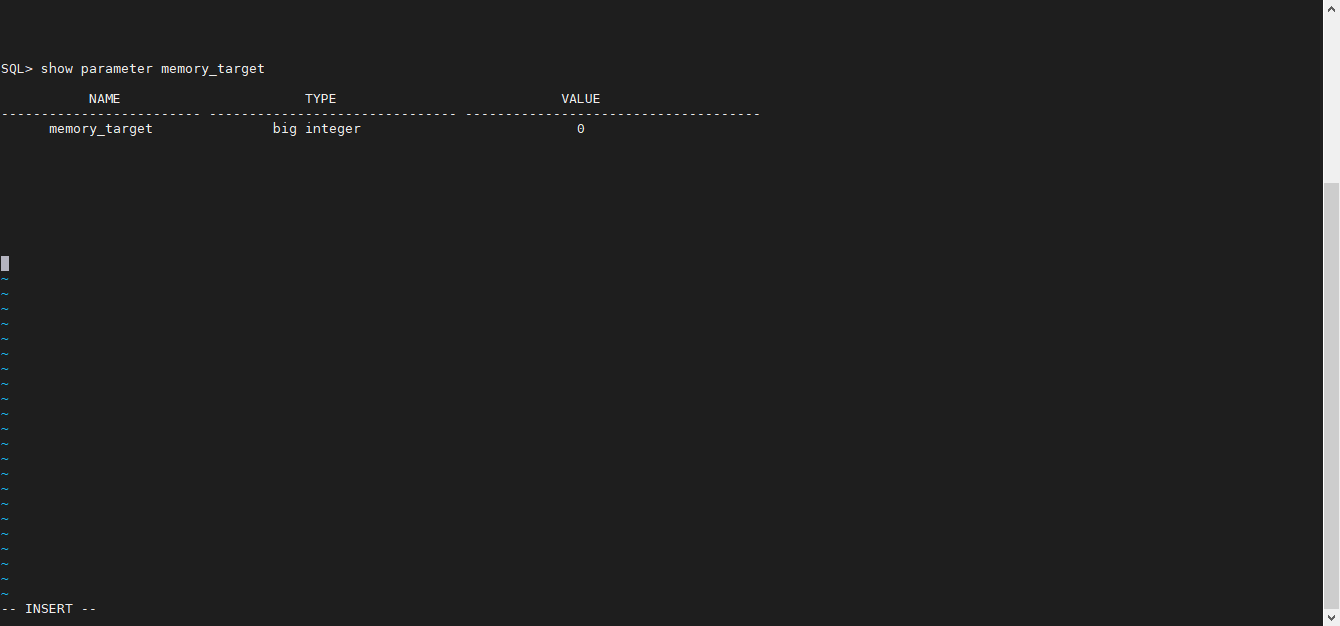
Alter system set memory\_target=1024M;

**说明：**

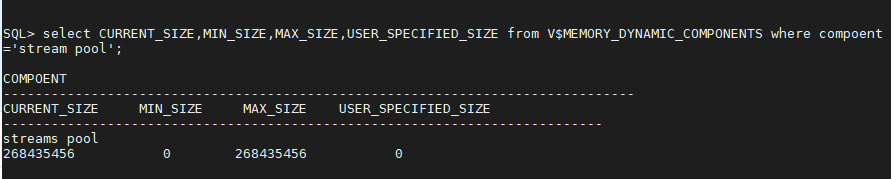
* 设置MEMORY\_MAX\_TARGET需要重启实例;
* 设置MEMORY\_TARGET的值不能超过MEMORY\_MAX\_TARGET;
* 查看指定参数值可以使用如下语句：

show parameter 参数名

如：查看MEMORY\_TARGET的值



* 当使用Automatic Memory Management时，仍然可以设置以下初始化参数：
* 如果SGA\_TARGET初始化参数也设置为非零值，则Automatic Memory Management将使用此值作为System Global Area(SGA)的最小值。
* 如果STREAMS\_POOL\_SIZE初始化参数也设置为非零值，则Automatic Memory Management会将此值用作Oracle Streams pool的最小值。
* 可以通过查询V$MEMORY\_DYNAMIC\_COMPONENTS视图来查看由Automatic Memory Management分配给Oracle Streams pool的当前内存。



* 使用Automatic Shared Memory Management设置Oracle Streams pool

满足以下条件时，Automatic Shared Memory Management将自动管理Oracle Streams pool的大小：

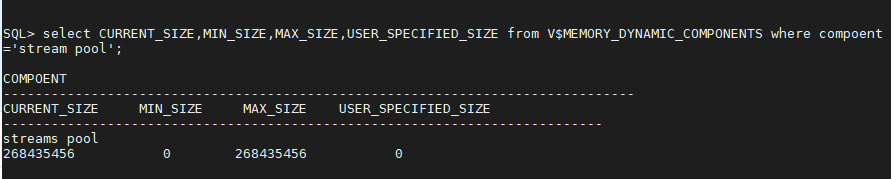
* 将MEMORY\_TARGET与MEMORY\_MAX\_TARGET初始化参数都设置为零。
* SGA\_TARGET 初始化参数设置为非零值。

如果使用Automatic Shared Memory Management，并且STREAMS\_POOL\_SIZE初始化参数也设置为非零值，则Automatic Shared Memory Management使用此值作为Oracle Streams pool的最小值。

可以通过查询V$MEMORY\_DYNAMIC\_COMPONENTS视图来查看由Automatic Shared Memory Management分配给Oracle Streams pool的当前内存。

select \* from V$MEMORY\_DYNAMIC\_COMPONENTS where component=’streams pool’;

示例如下：



**说明：**设置参数使用的语句： alter system set 参数名=参数值；

* 手动设置Oracle Streams pool

STREAMS\_POOL\_SIZE如果满足以下条件，则Oracle Streams pool大小是参数指定的值（以字节为单位）

**步骤1** MEMORY\_TARGET，MEMORY\_MAX\_TARGET和SGA\_TARGET初始化参数都设置为零.

**步骤2** STREAMS\_POOL\_SIZE初始化参数被设置为非零值。

如果计划手动设置Oracle Streams pool大小，则可以使用V$STREAMS\_POOL\_ADVICE动态性能视图来确定STREAMS\_POOL\_SIZE初始化参数的适当设置。

* 使用Oracle Streams pool的默认设置

如果MEMORY\_TARGET，MEMORY\_MAX\_TARGET，SGA\_TARGET和STREAMS\_POOL\_SIZE这些参数都设置为零，Oracle Streams pool大小将使用默认设置。默认情况下，如果设置了Oracle Streams池大小，则在数据库中首次使用Oracle Streams会将shared pool的10％的内存量从buffer cache转移到Oracle Streams pool。缓冲区高速缓存由DB\_CACHE\_SIZE初始化参数设置，共享池大小由SHARED\_POOL\_SIZE初始化参数设置。例如：

* DB\_CACHE\_SIZE 设置为 100 MB:

alter system set db\_cache\_size=100M;

* SHARED\_POOL\_SIZE设置为 80 MB:

alter system set shared\_pool\_size=80M;

* MEMORY\_TARGET, MEMORY\_MAX\_TARGET, SGA\_TARGE和STREAMS\_POOL\_SIZE都设置为零。

**说明：**设置MEMORY\_MAX\_TARGET后需要重启实例；

则最终的内存分布为：

buffer cache 为92MB

shared pool 为80MB

Oracle Streams pool 为8MB

#### 创建相关表

* Oracle正向增量功能的前置步骤需要在源库Oracle及目标库Vastbase G100、Vastbase E100、PostgreSQL、MySQL、openGauss、GaussDB(for openGauss)  创建增量相关表。
* Oracle到MySQL正向增量需要在源库Oracle及目标库MySQL创建相关表

**步骤1** 登录目标库，登录用户需要与后续exBase平台使用的目标库数据源连接用户保持一致。

**步骤2** 切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本。

**说明：**当目标库为MySQL时切换至MySQL数据源连接用户同名的database

drop table if exists public.kafkaoffset;

create table public.kafkaoffset (

jobid varchar (128) primary key,

topic varchar (64),

lastoffset bigint,

lastsuboffset bigint,

last\_scn\_number bigint,

scnnumber bigint,

transaction\_id text,

applytime timestamp,

updatetime timestamp

);

**步骤3** 目标库需要将public.kafkaoffset的查询、插入、更新的权限授予给后续exBase平台使用的目标库。

GRANT SELECT,INSERT,UPDATE ON public.kafkaoffset TO 目标库用户名;

## SQL Server到Vastbase G100正向

### 前提条件

* SQL Server到目标库正向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| SQL Server 2008、2008 R2、2012、2014 | Vastbase G100 V2.2 Build 5  Vastbase G100 V2.2 Build 10 |

### 操作步骤

#### SQL Server用户授权

* 增量功能需要在源库进行授权和操作，涉及与配置数据源相关信息如下图所示，根据实际填写情况调整。



#### SQL Server配置

**步骤1** 开启代理功能

* 查看SQL Server是否开启代理功能

执行如下sql查看代理是否开启成功。如结果为Running即为代理已开启；如结果为Stopped，可执行开启开启代理服务步骤。

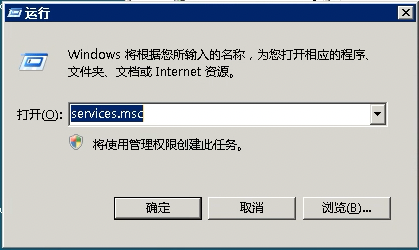
DECLARE @agent NVARCHAR(512);

SELECT @agent = COALESCE (N'SQLAgent$' + CONVERT (SYSNAME, SERVERPROPERTY ('InstanceName')), N'SQL ServerAgent');

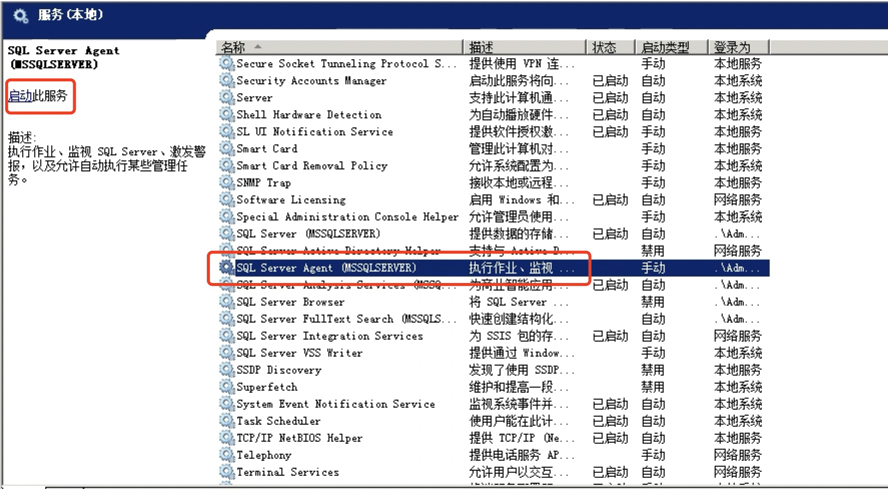
EXEC master.dbo.xp\_servicecontrol 'QueryState', @agent;

* 开启代理功能
* Windows版本

1、 按ctrl+r打开运行窗口，输入services.msc，按下回车。



2、找到SQL Server Agent选项，点击启动此服务，状态为已启动则代表该步骤成功。



* Linux版：

对于SQL Server 2019和SQL Server 2017 CU4及更高版本，只需启用SQL Server代理，无需安装单独的包。

1、找到bin目录，执行以下命令：

sudo /opt/mssql/bin/mssql-conf set sqlagent.enabled true

2、重启SQL Server服务。

sudo systemctl restart mssql-server

**步骤2** 对连接库开启cdc采集

1、在连接库执行命令开启cdc，连接库为后续exBase平台使用的SQL Server连接数据库。

exec sys.sp\_cdc\_enable\_db

2、执行该sql查看库是否开启成功。当查询结果为1时，表示已成功开启；为0时表示未开启。

select IS\_CDC\_ENABLED from sys.databases where NAME = '库名'

**说明：** 如需关闭cdc，可执行如下语句：

exec sys.sp\_cdc\_disable\_db

**步骤3**对需要采集的表开启cdc

* 在连接库上对单表开启cdc，连接库为后续exBase平台使用的SQL Server连接数据库：

EXEC sys.sp\_cdc\_enable\_table

@source\_schema='模式名',

@source\_name='表名',

@capture\_instance = NULL,

@role\_name = NULL,

@index\_name = NULL,

@captured\_column\_list = NULL,

@filegroup\_name = NULL

**说明：**脚本中<模式名>为需要开启cdc的表所属的模式名，<表名>为需要开启cdc的表名。

* 同一模式下所有表开启cdc：

1、在连接库上执行附件中存储过程：

EXEC sys.sp\_cdc\_enable\_table

@source\_schema='模式名',

@source\_name='表名',

@capture\_instance = NULL,

@role\_name = NULL,

@index\_name = NULL,

@captured\_column\_list = NULL,

@filegroup\_name = NULL

Exec sys.sp\_cdc\_disable\_table

@source\_schema='模式名',

@source\_name='表名',

@capture\_instance = 'all'

create procedure dbo.enable\_cdc @schema\_name varchar(50)

as

begin

declare @temp varchar(100)

declare tb\_cursor cursor

for (SELECT name as tableName FROM sys.tables WHERE schema\_id in

(select schema\_id from sys.schemas where name=@schema\_name))

open tb\_cursor

fetch next from tb\_cursor into @temp

while @@fetch\_status=0

begin

EXEC sys.sp\_cdc\_enable\_table

@source\_schema = @schema\_name,

@source\_name=@temp,

@role\_name= NULL

fetch next from tb\_cursor into @temp

end

close tb\_cursor

deallocate tb\_cursor

end

2、在连接库上对同一模式下所有表开启cdc：

exec enable\_cdc '模式名'

**说明：**脚本中的<模式名>为将开启/不开启cdc的表所属模式；

<表名>为需要开启/不开启cdc的表名。

* 检查表cdc是否开启成功。

SELECT is\_tracked\_by\_cdc, t1.name as tableName, t2.name as schemaName FROM sys.tables as t1, sys.schemas as t2 WHERE t1.schema\_id = t2.schema\_id and t1.name in ('表名1','表名2'...) and t2.name in ('模式名1','模式名2'...)

**说明：**当查询结果为1时，表示已成功开启；为0时表示未开启。

查询语句中<表名1/2>、<模式名1/2>为待查询是否开启cdc的模式1/2下的表1/2的名称。

#### 创建相关表

SQL Server正向增量功能的前置步骤需要在源库 SQL Server 及目标库 Vastbase G100 创建增量相关表。

**步骤1** 登录SQL Server数据库，登录用户需要与后续exBase平台使用的SQL Server数据源连接用户保持一致。

**步骤2** 切换至连接Sql Server数据源的database下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE SCHEMA ${username};

CREATE TABLE ${username}.incremental\_offset(

jobid varchar(128) NOT NULL,

topic varchar(64) NOT NULL,

origin\_startscn varchar(64) NULL,

changeLsn varchar(100) NULL,

commitLsn varchar(100) NULL,

transaction\_id text NULL,

collectoffset numeric(18,0) NULL,

sourcetime datetime NULL,

updatetime datetime DEFAULT getdate() NULL,

collecttime datetime NULL

)

**步骤3** 登录目标库，登录用户需要与后续exBase平台使用的目标库数据源连接用户保持一致。

**步骤4** 切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

drop table if exists public.kafkaoffset;

create table public.kafkaoffset (

jobid varchar (128) primary key,

topic varchar (64),

lastoffset bigint,

lastsuboffset bigint,

last\_scn\_number bigint,

scnnumber bigint,

transaction\_id text,

applytime timestamp,

updatetime timestamp,

commitposition text

);

**步骤5** 目标库需要将public.kafkaoffset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE ON public.kafkaoffset TO 目标库用户名;

## Vastbase G100到MySQL正向

### 前提条件

* Vastbase G100到MySQL开启正向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| Vastbase G100 V2.2 Build 5  Vastbase G100 V2.2 Build 10 | MySQL 5.6、5.7、8.0 |

* 增量功能需要在源库、目标库进行授权和操作，涉及与配置数据源相关信息如下图所示，根据实际填写情况调整。



* Vastbase在exBase的连接用户需要有replication权限。

授权语句：

alter user 用户名replication；

可通过如下语句检查如果结果是t，说明配置完成。

select userepl from pg\_user where usename='${username}';

* MySQL在exBase数据源连接管理中使用的MySQL用户需要有相应的权限。

授权语句：

GRANT SELECT,REPLICATION SLAVE,REPLICATION CLIENT ON \*.\* TO 'MySQL用户'@'%';

GRANT SELECT,INSERT,UPDATE ON MySQL用户.\* TO 'MySQL用户'@'%';

GRANT PROCESS ON \*.\* to 'MySQL用户'@'%';

FLUSH PRIVILEGES;

**说明：**授权语句中的<MySQL用户>指的是连接MySQL数据源时的用户名，根据实际情况进行修改。

### 操作步骤

#### 源库Vastbase G100配置

**步骤1**配置白名单

在pg\_hba.conf文件末尾新增：

host replication user ip/掩码 md5

其中：

* user为源库连接用户即exBase数据源配置的用户
* ip为exBase所在服务器ip，掩码可以为0；

例如，当exBase服务器为172.16.103.118，Vastbase G100连接数据源时的用户为test，则需要在pg\_hba.conf文件末尾新增如见下图。

**步骤2**修改Vastbase G100配置参数

Vastbase G100在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤3** 重启Vastbase G100数据库。

vb\_ctl restart

#### 创建相关表

**步骤1** 登录源库Vastbase G100，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。

切换至连接源库数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 源库Vastbase G100需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录目标库MySQL，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。

**步骤4** 新建名为MySQL数据源连接用户同名的数据库。

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE DATABASE ${username};

**步骤5** 切换至MySQL数据源连接用户同名的database执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE TABLE ${username}.`kafkaoffset` (

`jobid` varchar(128) NOT NULL,

`topic` varchar(64) DEFAULT NULL,

`lastoffset` mediumtext,

`lastsuboffset` mediumtext,

`last\_scn\_number` mediumtext,

`scnnumber` mediumtext,

`transaction\_id` varchar(128) DEFAULT NULL,

`applytime` timestamp(6) NULL DEFAULT NULL,

`updatetime` timestamp NULL DEFAULT NULL,

PRIMARY KEY (`jobid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

## Vastbase G100到Vastbase G100正向

### 前提条件

* Vastbase G100正向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| Vastbase G100 V2.2 Build 5、  Vastbase G100 V2.2 Build 10 | Vastbase G100 V2.2 Build 5、  Vastbase G100 V2.2 Build 10 |

#### 用户授权

* 增量功能需要在源库、目标库进行授权和操作，涉及与配置数据源相关信息如下图所示，根据实际填写情况调整。



* VastbaseG100源库、目标库在exBase的连接用户需要有replication权限。

授权语句：

alter user 用户名replication；

可通过如下语句检查如果结果是t，说明配置完成。

select userepl from pg\_user where usename='${username}';

#### 源库Vastbase G100配置

**步骤1** 配置白名单

在pg\_hba.conf文件末尾新增：

host replication user ip/掩码 md5

其中：

* user为数据源连接用户即exBase数据源配置的用户
* ip为exBase所在服务器ip，掩码可以为0；

例如，当exBase服务器为172.16.103.118，Vastbase G100连接数据源时的用户为test，则需要在pg\_hba.conf文件末尾新增如见下图。

**步骤2** 修改Vastbase G100配置参数

在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤3** 重启Vastbase G100数据库。

vb\_ctl restart

### 创建相关表

**步骤1** 登录源库Vastbase G100，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接源库Vastbase G100数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录目标库VastbaseG100，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。

**步骤4** 切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

drop table if exists public.kafkaoffset;

create table public.kafkaoffset (

jobid varchar (128) primary key,

topic varchar (64),

lastoffset bigint,

lastsuboffset bigint,

last\_scn\_number bigint,

scnnumber bigint,

transaction\_id text,

applytime timestamp,

updatetime timestamp,

commitposition text

);

## Vastbase G100到Kafka正向

### 前提条件

* Vastbase G100到Kafka正向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| Vastbase G100 V2.2 Build 5、  Vastbase G100 V2.2 Build 10 | Kafka单机/集群 |

### 操作步骤

#### Vastbase G100用户授权

* 增量功能需要在源库、目标库进行授权和操作，涉及与配置数据源相关信息如下图所示，根据实际填写情况调整。



* Vastbase在exBase的连接用户需要有replication权限。

授权语句：

alter user 用户名replication；

可通过如下语句检查如果结果是t，说明配置完成。

select userepl from pg\_user where usename='${username}';

#### Vastbase G100配置

* 配置白名单

在pg\_hba.conf文件末尾新增：

host replication user ip/掩码 md5

其中：

* user为数据源连接用户即exBase数据源配置的用户。
* ip为exBase所在服务器ip，掩码可以为0；

例如，当exBase服务器为172.16.103.118，Vastbase G100连接数据源时的用户为test，则需要在pg\_hba.conf文件末尾新增如见下图。

* 修改Vastbase G100配置参数

Vastbase G100在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤3** 重启Vastbase G100数据库。

vb\_ctl restart

#### Kafka配置

##### 操作说明

* 新建数据源

目标源kafka支持单机/集群，格式如下图所示：



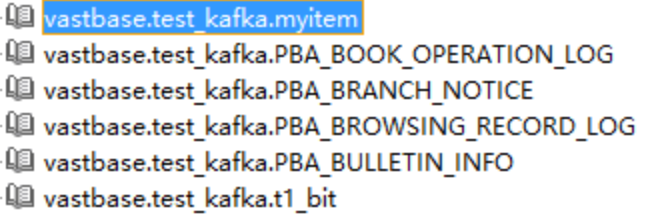
**说明：**集群环境地址ip:9092以逗号分割，数据源配置部分因支持集群的变动会影响之前的作业的同步，故之前的kafka数据源须重新创建。

* 新建作业A，选择需要增量同步的对象以及进行相关作业配置如区分大小写。
* 开启作业A增量迁移。
* 在Vastbase G100数据库中进行增删改操作。

##### 映射关系及消息格式声明

* 映射关系

以Kafka可视化工具Offset Explorer同步结果为例。



Topic与源库Vastbase G100的映射关系为：

{Databasename}.{Schemaname}.{Tablename}

* 消息格式

{

"sourceDatabaseName":"vastbase",

"commandType":"INSERT",

"objectOwner":"some\_schema",

"objectName":"some\_table",

"transaction":{

"totalOrder":"1",

"id":"xxx"

},

"oldColumnValues":{

"column\_name1":"column\_value1",

"column\_name2":"column\_name2",

"column\_name3":"column\_name3"

},

"newColumnValues":{

"column\_name1":"column\_value1",

"column\_name2":"column\_name2",

"column\_name3":"column\_name3"

},

"dataColumnType":{

"column\_name1":"column\_type1",

"column\_name2":"column\_type2",

"column\_name3":"column\_type3"

}

}

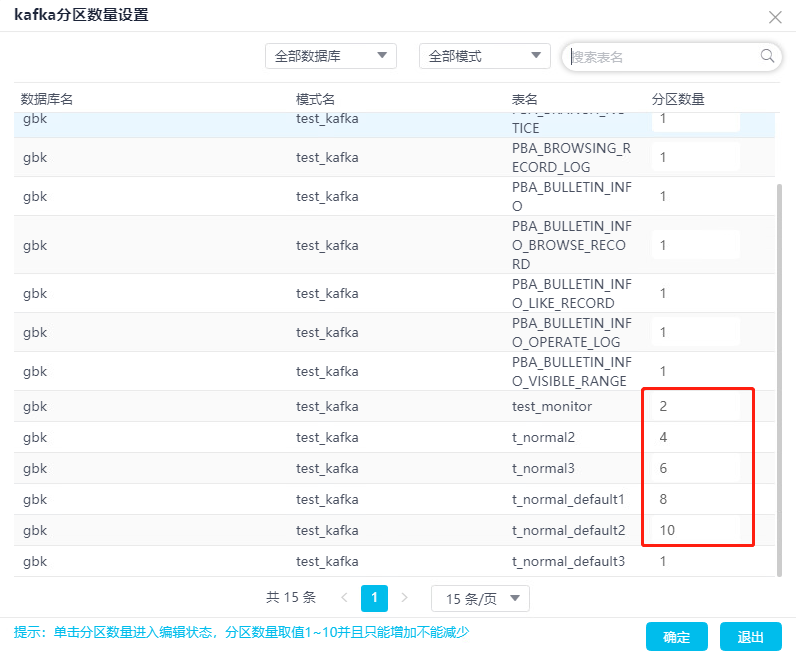
* sourceDatabaseName：源库名称。
* commandType：操作类型。一般是INSERT/UPDATE/DELETE/COMMIT。
* objectOwner：对象所属，一般是schema。
* objectName：对象名称，一般是表名。
* transaction：事务唯一标识。
* Transactionid：事务id。
* totalOrder：该记录是事务中的第几条记录。
* oldColumnValues：列的旧值。Key:Value为字段名column\_name:字段值column\_value,INSERT时为空数组。
* newColumnValues：列的新值。Key:Value为字段名column\_name:字段值column\_value,DELETE时为空数组。
* dataColumnType：列的数据类型。Key:Value为字段名column\_name：字段数据类型column\_type。

#### 作业配置

**步骤1** 新建正向增量作业A，源库选择Vastbase G100，目标库选择Kafka。选择要增量同步的所有对象。

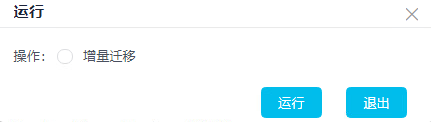
**步骤2** 进行相关作业配置，如是否区分大小写。

例如kafka分区数量设置，仅支持对主键表/复合主键表进行分区数量设置，分区范围为1-10，且修改分区数量时仅支持增加分区数量。



说明：对无主键普通表修改分区数不生效，修改分区数量须停止正在运行的作业，且分区数量变更仅对新数据生效，修改前已同步的数据保持不变。

**步骤3** 运行作业A的增量迁移环节，作业增量迁移状态显示为“运行中“。



**步骤4** 检查Vastbase G100数据库中的复制槽是否正常生成。

连接到Vastbase G100数据库，执行以下sql语句，检查复制槽是否正常生成。

select \* from pg\_catalog.pg\_replication\_slots where slot\_name like ‘%o2v111121%’;

注：此处 o2v111121 为作业B的名称，按实际情况进行填写。

sql语句查询结果有返回值，则表示复制槽正常生成。

**步骤5** 源库执行增删改操作，检查目标库增量同步情况。

### 注意事项

相关注意事项望知悉：

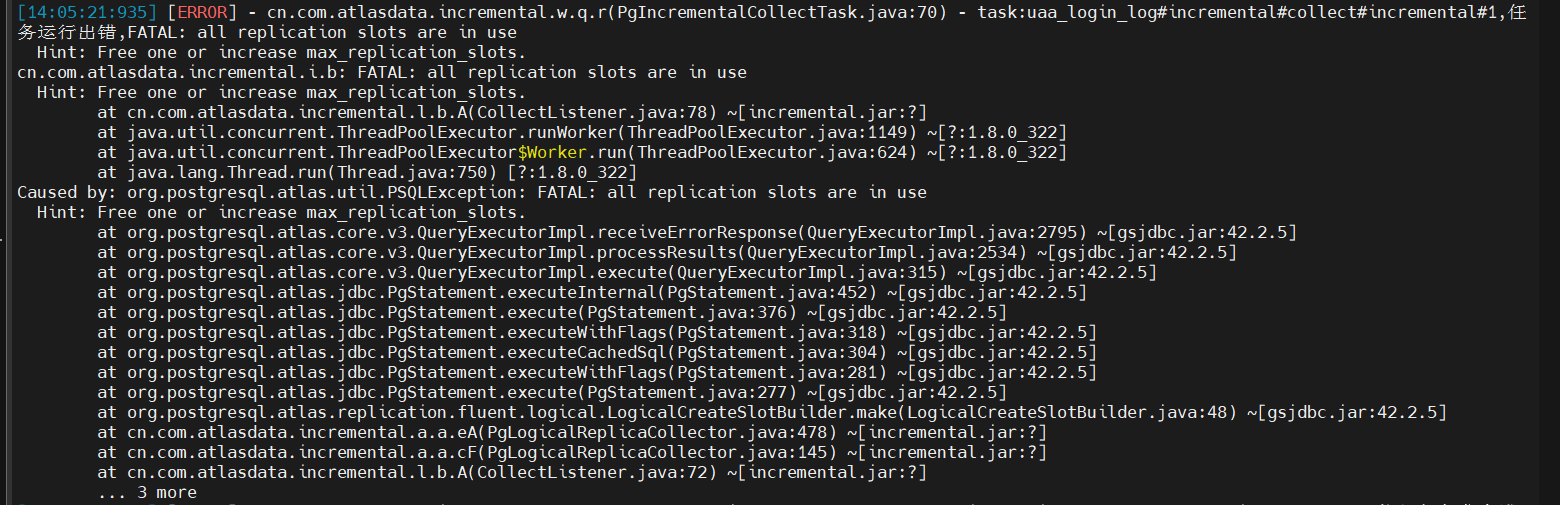
* Kafka不支持中文topic名，意即不支持带中文对象名的迁移。
* Kafka目前不支持大数据量的DML的迁移，具体边界值为单行数据量不超过20971520字节即20M左右。
* 消费者（调用API产生、自带shell程序产生、图形化工具产生等）也会创建topic，请注意区分。
* Vastbase G100 V2.2.5常见的数据类型均已支持，目前实测增量组件decodebuf在处理一个超出范围的时间(类型为timestamp，值为infinity)使用了一个错误的结构体来处理，会导致coredump。

### FAQ

#### Replication slot

* **复制槽占满（all replication slots are in use)**

**报错信息：**如图

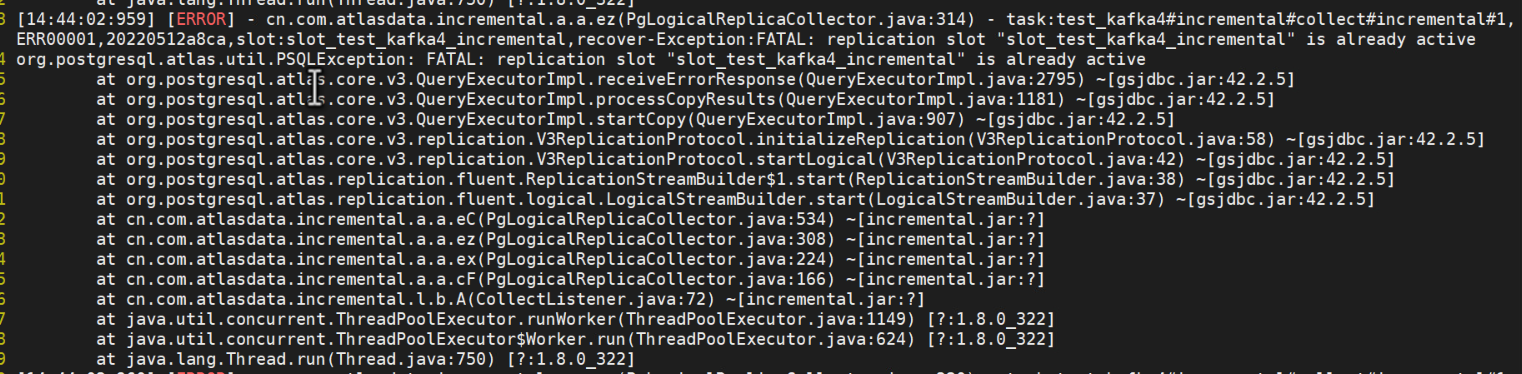


**报错原因：**源数据库的逻辑复制槽数量达到最大限制，exBase端无法新建复制槽。

**解决方案：**去源数据库修改/$PGDATA/postgresql.conf文件中的。

* **复制槽已处于活跃状态(replication slot is already active)**

**报错信息：**如图



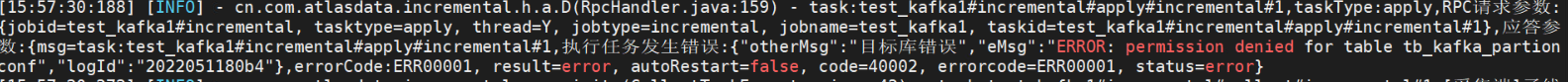
**报错原因：**wal\_sender\_timeout参数被注释或设置过低。

**解决方案：**去源数据库修改/$PGDATA/postgresql.conf文件中的wal\_sender\_timeout≥60s即可。

#### 数据库错误

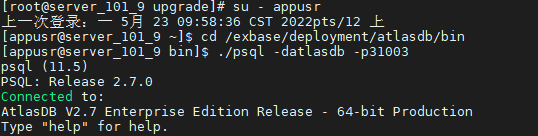
* **permission denied for table tb\_kafka\_partionconf**

**报错信息：**如图



**报错原因：**exBase配置库中exBase模式下该表tb\_kafka\_partionconf的owner错误。

**解决方案：**连接exBase配置库，并修改上表的owner为exBase，具体操作见下图：



# 反向增量迁移前置步骤

## MySQL到Vastbase G100反向

### 前提条件

* MySQL到Vastbase G100反向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| MySQL 5.5、5.6、5.7、8.0 | Vastbase G100 V2.2 Build 5、  Vastbase G100 V2.2 Build 10 |

* exBase版本支持情况如下：

|  |  |
| --- | --- |
| **平台** | **版本号** |
| exBase | 2.10及以上版本 |

### 操作步骤

#### 用户授权

* 目标库Vastbase G100在exBase配置数据源时使用的用户需要有replication的权限。授权语句如下：

alter user 目标库用户名 replication;

* 需要把public模式的权限赋予连接用户：

grant all on schema public to 目标库用户名;

**说明：**授权语句中的<目标库用户名>指的是添加目标库数据源时使用的用户名。

#### 目标库配置

**步骤1** 配置白名单

在目标库Vastbase G100配置应用端IP的复制权限，在pg\_hba.conf文件末尾新增一行：

host replication user ip/掩码 md5

其中：

* user为exBase配置数据源连接时使用的用户；
* ip为exBase所在服务器ip，掩码可以为0；

例如，当exBase服务器为172.16.103.118，Vastbase G100连接数据源时的用户为test，则需要在pg\_hba.conf文件末尾新增如见下图。

**步骤2** 修改Vastbase G100配置参数

目标库Vastbase G100在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤3** 重启Vastbase G100数据库。

vb\_ctl restart

#### 创建表

**步骤1** 登录目标库Vastbase G100，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 目标库需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录源库MySQL，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至MySQL数据源连接用户同名的database执行以下sql脚本：

**说明：**反向增量的前提是MySQL到Vastbase G100正向增量时已创建了与MySQL数据源连接用户同名的数据库。

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE TABLE ${username}.`kafkaoffset` (

`jobid` varchar(128) NOT NULL,

`topic` varchar(64) DEFAULT NULL,

`lastoffset` mediumtext,

`lastsuboffset` mediumtext,

`last\_scn\_number` mediumtext,

`scnnumber` mediumtext,

`transaction\_id` varchar(128) DEFAULT NULL,

`applytime` timestamp(6) NULL DEFAULT NULL,

`updatetime` timestamp NULL DEFAULT NULL,

PRIMARY KEY (`jobid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

## MySQL到Vastbase E100反向

### 前提条件

* MySQL到Vastbase E100/PostgreSQL反向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| MySQL 5.6、5.7、8.0 | Vastbase E100 V3.0 |

* exBase版本支持情况如下：

|  |  |
| --- | --- |
| **平台** | **版本号** |
| exBase | 2.10及以上版本 |

### 操作步骤

#### 用户授权

* 目标库Vastbase E100在exBase配置数据源时使用的用户需要有replication的权限。授权语句如下：

alter user 目标库用户名 replication;

* 需要把public模式的权限赋予连接用户：

grant all on schema public to 目标库用户名;

**说明：**授权语句中的<目标库用户名>指的是添加目标库数据源时使用的用户名。

#### 安装插件

若需要使用反向增量迁移功能，需要在Vastbase E100配置decoderbufs。

* 安装decoderbufs插件所用到的包：protobuf-master.zip、protobuf-c.zip、decoderbufs.zip。
* 安装前可以查下是否已有1.2+版本的protoc，如果有可以直接看步骤2的环境变量设置，然后从步骤3开始:

protoc --version

**步骤1** 安装protobuf

1. 安装protobuf依赖：

yum install autoconf

yum install automake

yum install libtool

yum install make

yum install gcc gcc-c++

yum install unzip

1. 编译安装protobuf :

unzip protobuf-master.zip

cd protobuf-master

./autogen.sh

./configure --prefix=/usr

make

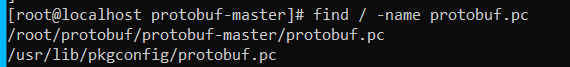
make install

ldconfig

**步骤2** 安装protobuf-c

(1)设置环境变量PKG\_CONFIG\_PATH

确认PKG\_CONFIG\_PATH包含protobuf.pc文件父目录，对应步骤1里的protobuf的安装目录下的pkgconfig。



export PKG\_CONFIG\_PATH=/usr/lib/pkgconfig:$PKG\_CONFIG\_PATH

(2)编译安装protobuf-c

unzip protobuf-c.zip

cd protobuf-c

chmod u+x autogen.sh

./autogen.sh

./configure --prefix=/usr

make

make install

(3)设置环境变量LD\_LIB\_RARY\_PATH

确认LD\_LIBRARY\_PATH包含 libprotobuf-c.so的父目录, 即对应protobuf-c的安装目录：

export LD\_LIBRARY\_PATH=/usr/lib:$LD\_LIBRARY\_PATH

**说明：**(1)(3) 设置的环境变量也可以写入~/.bashrc 或者/etc/profile等文件持久化。

**步骤3** 编译安装decoderbufs

unzip postgres-decoderbufs.zip

cd postgres-decoderbufs

export PATH=$PGHOME/bin:$PATH

make

make install

PGHOME指数据库的安装目录。

如需要为其它的同一版本的数据库安装，可以复制使用同版本数据库编译生成的插件，在当前目录分别复制编译生成的decoderbufs.so 和 decoderbufs.control 文件至以下对应目录：

mv decoderbufs.so $PGHOME/lib/postgresql

mv decoderbufs.control $PGHOME/share/postgresql/extension

若没有postgresql目录，统一去掉postgresql子目录:

mv decoderbufs.so $PGHOME/lib/

mv decoderbufs.control $PGHOME/share/extension

将两个文件的用户权限修改为数据库安装用户权限。

#### 目标库配置

**步骤1** 修改数据库配置

目标库在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤2** 重启数据库。

启动前注意启动用户的环境变量LD\_LIBRARY\_PATH需要包含 libprotobuf-c.so的父目录。

#### 创建表

**步骤1** 登录目标库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 目标库需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录MySQL源库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至MySQL数据源连接用户同名的database执行以下sql脚本

**说明：**反向增量的前提是MySQL到Vastbase E100正向增量时已创建了与MySQL数据源连接用户同名的数据库。

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE TABLE ${username}.`kafkaoffset` (

`jobid` varchar(128) NOT NULL,

`topic` varchar(64) DEFAULT NULL,

`lastoffset` mediumtext,

`lastsuboffset` mediumtext,

`last\_scn\_number` mediumtext,

`scnnumber` mediumtext,

`transaction\_id` varchar(128) DEFAULT NULL,

`applytime` timestamp(6) NULL DEFAULT NULL,

`updatetime` timestamp NULL DEFAULT NULL,

PRIMARY KEY (`jobid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

## MySQL到PostgreSQL反向

### 前提条件

* MySQL到PostgreSQL反向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| MySQL 5.6、5.7、8.0 | PostgreSQL 11g、12c |

* exBase版本支持情况如下：

|  |  |
| --- | --- |
| **平台** | **版本号** |
| exBase | 2.10及以上版本 |

### 操作步骤

#### 用户授权

* 目标库PostgreSQL在exBase配置数据源时使用的用户需要有replication的权限。授权语句如下：

alter user 目标库用户名 replication;

* 需要把public模式的权限赋予连接用户：

grant all on schema public to 目标库用户名;

**说明：**授权语句中的<目标库用户名>指的是添加目标库数据源时使用的用户名。

#### 安装插件

若需要使用反向增量迁移功能，需要在PostgreSQL配置decoderbufs。

* 安装decoderbufs插件所用到的包：protobuf-master.zip、protobuf-c.zip、decoderbufs.zip。
* 安装前可以查下是否已有1.2+版本的protoc，如果有可以直接看步骤2的环境变量设置，然后从步骤3开始:

protoc --version

**步骤1** 安装protobuf

1. 安装protobuf依赖：

yum install autoconf

yum install automake

yum install libtool

yum install make

yum install gcc gcc-c++

yum install unzip

1. 编译安装protobuf :

unzip protobuf-master.zip

cd protobuf-master

./autogen.sh

./configure --prefix=/usr

make

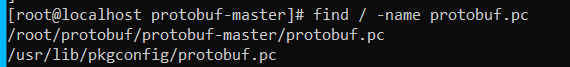
make install

ldconfig

**步骤2** 安装protobuf-c

(1)设置环境变量PKG\_CONFIG\_PATH

确认PKG\_CONFIG\_PATH包含protobuf.pc文件父目录，对应步骤1里的protobuf的安装目录下的pkgconfig。



export PKG\_CONFIG\_PATH=/usr/lib/pkgconfig:$PKG\_CONFIG\_PATH

(2)编译安装protobuf-c

unzip protobuf-c.zip

cd protobuf-c

chmod u+x autogen.sh

./autogen.sh

./configure --prefix=/usr

make

make install

(3)设置环境变量LD\_LIB\_RARY\_PATH

确认LD\_LIBRARY\_PATH包含 libprotobuf-c.so的父目录, 即对应protobuf-c的安装目录：

export LD\_LIBRARY\_PATH=/usr/lib:$LD\_LIBRARY\_PATH

**说明：**(1)(3) 设置的环境变量也可以写入~/.bashrc 或者/etc/profile等文件持久化。

**步骤3** 编译安装decoderbufs

unzip postgres-decoderbufs.zip

cd postgres-decoderbufs

export PATH=$PGHOME/bin:$PATH

make

make install

PGHOME指数据库的安装目录。

如需要为其它的同一版本的数据库安装，可以复制使用同版本数据库编译生成的插件，在当前目录分别复制编译生成的decoderbufs.so 和 decoderbufs.control 文件至以下对应目录：

mv decoderbufs.so $PGHOME/lib/postgresql

mv decoderbufs.control $PGHOME/share/postgresql/extension

若没有postgresql目录，统一去掉postgresql子目录:

mv decoderbufs.so $PGHOME/lib/

mv decoderbufs.control $PGHOME/share/extension

将两个文件的用户权限修改为数据库安装用户权限。

#### 目标库配置

**步骤1** 修改数据库配置

目标库在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤2** 重启数据库。

启动前注意启动用户的环境变量LD\_LIBRARY\_PATH需要包含 libprotobuf-c.so的父目录。

#### 创建表

**步骤1** 登录目标库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 目标库需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录MySQL源库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至MySQL数据源连接用户同名的database执行以下sql脚本：

**说明：**反向增量的前提是MySQL到Vastbase E100、PostgreSQL正向增量时已创建了与MySQL数据源连接用户同名的数据库。

-- ${username}请替换为exBase页面数据源配置的源库的username

CREATE TABLE ${username}.`kafkaoffset` (

`jobid` varchar(128) NOT NULL,

`topic` varchar(64) DEFAULT NULL,

`lastoffset` mediumtext,

`lastsuboffset` mediumtext,

`last\_scn\_number` mediumtext,

`scnnumber` mediumtext,

`transaction\_id` varchar(128) DEFAULT NULL,

`applytime` timestamp(6) NULL DEFAULT NULL,

`updatetime` timestamp NULL DEFAULT NULL,

PRIMARY KEY (`jobid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

## Oracle到Vastbase G100反向

### 前提条件

* Oracle到VastbaseG100反向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| Oracle 11g（11.2.0.4以上）  Oracle 12c（支持CDB）  Oracle 18c  Oracle 19c | Vastbase G100 V2.2 Build 5、V2.2 Build 10 |
| Oracle 12c（支持PDB） | Vastbase G100 V2.2 Build 5、V2.2 Build 10 |

* exBase版本支持情况如下：

|  |  |
| --- | --- |
| **平台** | **版本号** |
| exBase | 2.10及以上版本 |

### 操作步骤

#### 用户授权

* 目标库Vastbase G100、Vastbase E100、PostgreSQL在exBase配置数据源时使用的用户需要有replication的权限。  
  授权语句如下：

alter user 目标库用户名 replication;

* 若Vastbase G100作采集库（如Vastbase G100正向增量、Oracle/MySQL to Vastbase G10的反向增量），则需要把PUBLIC这一schema的权限赋予连接用户：

grant all on schema public to 目标库用户名;

**说明：**授权语句中的<目标库用户名>指的是添加目标库数据源时使用的用户名。

#### 目标库配置

**步骤1** 配置白名单

在目标库Vastbase G100配置应用端IP的复制权限，在pg\_hba.conf文件末尾新增一行：

host replication user ip/掩码 md5

其中：

* user为exBase配置数据源连接时使用的用户；
* ip为exBase所在服务器ip，掩码可以为0；

例如，当exBase服务器为172.16.103.118，Vastbase G100连接数据源时的用户为test，则需要在pg\_hba.conf文件末尾新增如见下图。

**步骤2**修改Vastbase G100配置参数

目标库Vastbase G100在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤3** 重启Vastbase G100数据库。

vb\_ctl restart

#### 创建表

**步骤1** 登录Vastbase G100目标库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 目标库需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录源库Oracle，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接Oracle数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

drop table ${username}.kafkaoffset;

create table ${username}.kafkaoffset (

jobid varchar2 (128) primary key,

topic varchar2 (64),

lastoffset number,

lastsuboffset number,

last\_scn\_number number,

scnnumber number,

transaction\_id clob,

applytime timestamp,

updatetime timestamp

);

## Oracle到Vastbase E100反向

### 前提条件

* Oracle到Vastbase E100反向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| Oracle 11g（11.2.0.4以上）  Oracle 12c（支持CDB）  Oracle 18c  Oracle 19c | Vastbase E100 V3.0 |

* exBase版本支持情况如下：

|  |  |
| --- | --- |
| **平台** | **版本号** |
| exBase | 2.10及以上版本 |

### 操作步骤

#### 用户授权

* 目标库Vastbase E100在exBase配置数据源时使用的用户需要有replication的权限。授权语句如下：

alter user 目标库用户名 replication;

* 需要把public模式的权限赋予连接用户：

grant all on schema public to 目标库用户名;

**说明：**授权语句中的<目标库用户名>指的是添加目标库数据源时使用的用户名。

#### 安装插件

若需要使用反向增量迁移功能，需要在Vastbase E100配置decoderbufs。

* 安装decoderbufs插件所用到的包：protobuf-master.zip、protobuf-c.zip、decoderbufs.zip。
* 安装前可以查下是否已有1.2+版本的protoc，如果有可以直接看步骤2的环境变量设置，然后从步骤3开始:

protoc --version

**步骤1** 安装protobuf

1. 安装protobuf依赖：

yum install autoconf

yum install automake

yum install libtool

yum install make

yum install gcc gcc-c++

yum install unzip

1. 编译安装protobuf :

unzip protobuf-master.zip

cd protobuf-master

./autogen.sh

./configure --prefix=/usr

make

make install

ldconfig

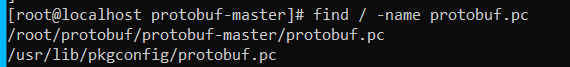
**步骤2** 安装protobuf-c

(1)设置环境变量PKG\_CONFIG\_PATH

确认PKG\_CONFIG\_PATH包含protobuf.pc文件父目录，对应步骤1里的protobuf的安装目录下的pkgconfig。

export PKG\_CONFIG\_PATH=/usr/lib/pkgconfig:$PKG\_CONFIG\_PATH

**说明：**可通过如下示例确定目录



(2)编译安装protobuf-c

unzip protobuf-c.zip

cd protobuf-c

chmod u+x autogen.sh

./autogen.sh

./configure --prefix=/usr

make

make install

(3)设置环境变量LD\_LIB\_RARY\_PATH

确认LD\_LIBRARY\_PATH包含 libprotobuf-c.so的父目录, 即对应protobuf-c的安装目录：

export LD\_LIBRARY\_PATH=/usr/lib:$LD\_LIBRARY\_PATH

**说明：**(1)(3) 设置的环境变量也可以写入~/.bashrc 或者/etc/profile等文件持久化。

**步骤3** 编译安装decoderbufs

unzip postgres-decoderbufs.zip

cd postgres-decoderbufs

export PATH=$PGHOME/bin:$PATH

make

make install

PGHOME指数据库的安装目录。

如需要为其它的同一版本的数据库安装，可以复制使用同版本数据库编译生成的插件，在当前目录分别复制编译生成的decoderbufs.so 和 decoderbufs.control 文件至以下对应目录：

mv decoderbufs.so $PGHOME/lib/postgresql

mv decoderbufs.control $PGHOME/share/postgresql/extension

若没有postgresql目录，统一去掉postgresql子目录:

mv decoderbufs.so $PGHOME/lib/

mv decoderbufs.control $PGHOME/share/extension

将两个文件的用户权限修改为数据库安装用户权限。

#### 目标库配置

**步骤1** 修改数据库配置

目标库在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。



其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤2** 重启数据库。

启动前注意启动用户的环境变量LD\_LIBRARY\_PATH需要包含 libprotobuf-c.so的父目录。

#### 创建表

**步骤1** 登录目标库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。

切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 目标库需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录Oracle源库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接Oracle数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

drop table ${username}.kafkaoffset;

create table ${username}.kafkaoffset (

jobid varchar2 (128) primary key,

topic varchar2 (64),

lastoffset number,

lastsuboffset number,

last\_scn\_number number,

scnnumber number,

transaction\_id clob,

applytime timestamp,

updatetime timestamp

);

## Oracle到PostgreSQL反向

### 前提条件

* Oracle到PostgreSQL反向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| Oracle 11g（11.2.0.4以上）  Oracle 12c（支持CDB）  Oracle 18c  Oracle 19c | PostgreSQL 11g、12c |

* exBase版本支持情况如下：

|  |  |
| --- | --- |
| **平台** | **版本号** |
| exBase | 2.10及以上版本 |

### 操作步骤

#### 用户授权

* 目标库PostgreSQL在exBase配置数据源时使用的用户需要有replication的权限。授权语句如下：

alter user 目标库用户名 replication;

* 需要把public模式的权限赋予连接用户：

grant all on schema public to 目标库用户名;

**说明：**授权语句中的<目标库用户名>指的是添加目标库数据源时使用的用户名。

#### 安装插件

若需要使用反向增量迁移功能，需要在PostgreSQL配置decoderbufs。

* 安装decoderbufs插件所用到的包：protobuf-master.zip、protobuf-c.zip、decoderbufs.zip。
* 安装前可以查下是否已有1.2+版本的protoc，如果有可以直接看步骤2的环境变量设置，然后从步骤3开始:

protoc --version

**步骤1** 安装protobuf

1. 安装protobuf依赖：

yum install autoconf

yum install automake

yum install libtool

yum install make

yum install gcc gcc-c++

yum install unzip

1. 编译安装protobuf :

unzip protobuf-master.zip

cd protobuf-master

./autogen.sh

./configure --prefix=/usr

make

make install

ldconfig

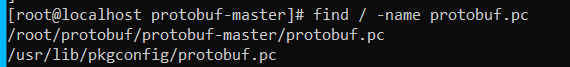
**步骤2** 安装protobuf-c

(1)设置环境变量PKG\_CONFIG\_PATH

确认PKG\_CONFIG\_PATH包含protobuf.pc文件父目录，对应步骤1里的protobuf的安装目录下的pkgconfig。

export PKG\_CONFIG\_PATH=/usr/lib/pkgconfig:$PKG\_CONFIG\_PATH

**说明：**可通过如下示例确定目录



(2)编译安装protobuf-c

unzip protobuf-c.zip

cd protobuf-c

chmod u+x autogen.sh

./autogen.sh

./configure --prefix=/usr

make

make install

(3)设置环境变量LD\_LIB\_RARY\_PATH

确认LD\_LIBRARY\_PATH包含 libprotobuf-c.so的父目录, 即对应protobuf-c的安装目录：

export LD\_LIBRARY\_PATH=/usr/lib:$LD\_LIBRARY\_PATH

**说明：**(1)(3) 设置的环境变量也可以写入~/.bashrc 或者/etc/profile等文件持久化。

**步骤3** 编译安装decoderbufs

unzip postgres-decoderbufs.zip

cd postgres-decoderbufs

export PATH=$PGHOME/bin:$PATH

make

make install

PGHOME指数据库的安装目录。

如需要为其它的同一版本的数据库安装，可以复制使用同版本数据库编译生成的插件，在当前目录分别复制编译生成的decoderbufs.so 和 decoderbufs.control 文件至以下对应目录：

mv decoderbufs.so $PGHOME/lib/postgresql

mv decoderbufs.control $PGHOME/share/postgresql/extension

若没有postgresql目录，统一去掉postgresql子目录:

mv decoderbufs.so $PGHOME/lib/

mv decoderbufs.control $PGHOME/share/extension

将两个文件的用户权限修改为数据库安装用户权限。

#### 目标库配置

**步骤1** 修改数据库配置

目标库在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。



其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤2** 重启数据库。

启动前注意启动用户的环境变量LD\_LIBRARY\_PATH需要包含 libprotobuf-c.so的父目录。

#### 创建表

**步骤1** 登录目标库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。

切换至连接目标库数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

create table ${username}.incremental\_offset (

jobid varchar(128) primary key,

topic varchar(64) not null,

startlsn bigint,

collectcommitlsn bigint,

collectoffset bigint,

slotname varchar(128),

sourcetime timestamp,

updatetime timestamp default current\_timestamp,

collecttime timestamp

);

**步骤2** 目标库需要将${username}.incremental\_offset的查询、插入、更新的权限授予给后续exBase平台使用的目标库数据源连接用户。

GRANT SELECT,INSERT,UPDATE on ${username}.incremental\_offset to ${username}；

**步骤3** 登录Oracle源库，登录用户需要与后续exBase平台使用的数据源连接用户保持一致。切换至连接Oracle数据源的database、数据源连接用户的schema下执行以下sql脚本：

-- ${username}请替换为exBase页面数据源配置的源库的username

drop table ${username}.kafkaoffset;

create table ${username}.kafkaoffset (

jobid varchar2 (128) primary key,

topic varchar2 (64),

lastoffset number,

lastsuboffset number,

last\_scn\_number number,

scnnumber number,

transaction\_id clob,

applytime timestamp,

updatetime timestamp

);

## SQL Server到Vastbase G100反向

### 前提条件

* SQL Server到Vastbase G100反向增量迁移版本支持情况如下：

| **源库** | **目标库** |
| --- | --- |
| SQL Server 2005、2008、2008 R2、2012、2014 | Vastbase G100 V2.2 Build 5、  Vastbase G100 V2.2 Build 10 |

* exBase版本支持情况如下：

|  |  |
| --- | --- |
| **平台** | **版本号** |
| exBase | 2.10及以上版本 |

### 操作步骤

#### 用户授权

* 目标库Vastbase G100、Vastbase E100、PostgreSQL在exBase配置数据源时使用的用户需要有replication的权限。  
  授权语句如下：

alter user 目标库用户名 replication;

* 若Vastbase G100作采集库（如Vastbase G100正向增量、Oracle/MySQL to Vastbase G10的反向增量），则需要把PUBLIC这一schema的权限赋予连接用户：

grant all on schema public to 目标库用户名;

**说明：**授权语句中的<目标库用户名>指的是添加目标库数据源时使用的用户名。

#### 目标库配置

**步骤1** 配置白名单

在目标库Vastbase G100配置应用端IP的复制权限，在pg\_hba.conf文件末尾新增一行：

host replication user ip/掩码 md5

其中：

* user为exBase配置数据源连接时使用的用户；
* ip为exBase所在服务器ip，掩码可以为0。

例如，当exBase服务器为172.16.103.118，Vastbase G100连接数据源时的用户为test，则需要在pg\_hba.conf文件末尾新增如见下图。

**步骤2** 修改数据库配置

目标库Vastbase G100在postgresql.conf配置文件中进行相关配置。

* wal\_level 改为logical；
* shared\_preload\_libraries 选项添加 decoderbufs ；
* 根据需要调整max\_wal\_senders、max\_replication\_slots和wal\_sender\_timeout的值。





其中：

* max\_wal\_senders：默认值为4。由于一个作业要占一个slot，则sender要大于等于max\_replication\_slots值。
* max\_replication\_slots：默认值是10 ，则表示最多只能建立10个复制槽
* wal\_sender\_timeout：默认值为6s，建议值为50s，过小可能导致超时异常的报错。

**步骤3** 重启Vastbase G100数据库。

vb\_ctl restart



电话：010-82838118

地址：北京市海淀区学院路30号科大天工大厦B座6层

官网：www.vastdata.com.cn