

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
1.1. MySQL 到目标库正向（通用）	1
1.1.1. 前提条件	1
1.1.2. 操作步骤	1
1.2. Oracle 到目标库正向（通用）	5
1.2.1. 前提条件	5
1.2.2. 操作步骤	5
1.3. SQL Server 到 Vastbase G100 正向	13
1.3.1. 前提条件	13
1.3.2. 操作步骤	13
1.4. Vastbase G100 到 MySQL 正向	18
1.4.1. 前提条件	18
1.4.2. 操作步骤	20
1.5. Vastbase G100 到 Vastbase G100 正向	22
1.5.1. 前提条件	22
1.5.2. 创建相关表	24
1.6. Vastbase G100 到 Kafka 正向	25
1.6.1. 前提条件	25
1.6.2. 操作步骤	25
1.6.3. 注意事项	30
1.6.4. FAQ	30
2. 反向增量迁移前置步骤	31
2.1. MySQL 到 Vastbase G100 反向	31
2.1.1. 前提条件	31
2.1.2. 操作步骤	32
2.2. MySQL 到 Vastbase E100 反向	34
2.2.1. 前提条件	34
2.2.2. 操作步骤	35
2.3. MySQL 到 PostgreSQL 反向	39
2.3.1. 前提条件	39
2.3.2. 操作步骤	39
2.4. Oracle 到 Vastbase G100 反向	43
2.4.1. 前提条件	43
2.4.2. 操作步骤	44
2.5. Oracle 到 Vastbase E100 反向	46

2.5.1. 前提条件	46
2.5.2. 操作步骤	46
2.6. Oracle 到 PostgreSQL 反向	50
2.6.1. 前提条件	50
2.6.2. 操作步骤	51
2.7. SQL Server 到 Vastbase G100 反向	55
2.7.1. 前提条件	55
2.7.2. 操作步骤	55

1. 正向增量迁移前置步骤

1.1. 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。

1.1.1. 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

1.1.2. 操作步骤

1.1.2.1. 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。

1.1.2.2. 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：

```
28 #character-set-server=utf8mb4
29 #symbolic-links=0
30 #explicit_defaults_for_timestamp=true
31 [mysqld]
32 user=root
33 datadir=/usr/local/mysql/data
34 basedir=/usr/local/mysql
35 port=3306
36 max_connections=200
37 max_connect_errors=10
38 character-set-server=utf8
39 default-storage-engine=INNODB
40 default_authentication_plugin=mysql_native_password
41 lower_case_table_names=1
42 group_concat_max_len=102400
43 log-bin=mysql-bin
44 binlog-format=ROW
45 service_id=1
46 [mysql]
47 default-character-set=utf8
48 [client]
49 port=3306
50 default-character-set=utf8

INSERT my.cnf[+]
-- INSERT --
```

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

```
[root@CentOS76-53 etc]# service mysql restart
Shutting down MySQL.. SUCCESS!
Starting MySQL.. SUCCESS!
[root@CentOS76-53 etc]#
```

1.1.2.3. 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,
`fullmigratetimestamp` 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;
```

1.2. 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、Gauss DB(for openGauss) 时前置操作一致，因此本小节内容通用。

1.2.1. 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

1.2.2. 操作步骤

1.2.2.1. 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

1.2.2.2. 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

```
SQL> show parameter memory_target
```

NAME	TYPE	VALUE
memory_target	big integer	0

- 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。

```
SQL> select CURRENT_SIZE,MIN_SIZE,MAX_SIZE,USER_SPECIFIED_SIZE from V$MEMORY_DYNAMIC_COMPONENTS where component='stream pool';
```

COMPONENT	CURRENT_SIZE	MIN_SIZE	MAX_SIZE	USER_SPECIFIED_SIZE
streams pool	268435456	0	268435456	0

- 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：

```
SQL> select CURRENT_SIZE,MIN_SIZE,MAX_SIZE,USER_SPECIFIED_SIZE from V$MEMORY_DYNAMIC_COMPONENTS where component='stream pool';
```

COMPONENT	CURRENT_SIZE	MIN_SIZE	MAX_SIZE	USER_SPECIFIED_SIZE
streams pool	268435456	0	268435456	0

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_TARGET 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

1.2.2.3. 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;
```

1.3. SQL Server 到 Vastbase G100 正向

1.3.1. 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

1.3.2. 操作步骤

1.3.2.1. 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。

1.3.2.2. 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 FR
OM sys.tables as t1, sys.schemas as t2 WHERE t1.schema_id = t2.schema_id an
d 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。

1.3.2.3. 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;
```

1.4. Vastbase G100 到 MySQL 正向

1.4.1. 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 username='${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。

1.4.2. 操作步骤

1.4.2.1. 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。

```
host replication test 172.16.103.110/0 md5
"pg_hba.conf" 100L,4526c
```

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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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
```

1.4.2.2. 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;

```

1.5.Vastbase G100 到 Vastbase G100 正向

1.5.1. 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

1.5.1.1. 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 username='${username}';
```

1.5.1.2. 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。

```
host replication test 172.16.103.110/0 md5
"pg_hba.conf" 100L,4526c
```

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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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
```

1.5.2. 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
);

```

1.6. Vastbase G100 到 Kafka 正向

1.6.1. 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

1.6.2. 操作步骤

1.6.2.1. 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。

连接配置

* 数据源名称: G100_123

* 类型: Vastbase G100

* 数据库名: vastbase 数据源连接database

* IP地址: 172.16.105.60

* 端口: 5432

* 应用环境字符集: GBK

* 用户名: test 数据源连接用户

* 密码:

* 是否为集群: ☐ 是 ☒ 否

测试 保存 退出

❖ 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 username='${username}';
```

1.6.2.2. 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。

```
host replication test 172.16.103.110/0 md5
"pg_hba.conf" 100L,4526c
```

❖ 修改 Vastbase G100 配置参数

Vastbase G100 在 postgresql.conf 配置文件中相关配置。

- ❖ 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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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
```

1.6.2.3. Kafka 配置

1.6.2.3.1.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：



连接配置

* 数据源名称: kafka_cluster

* 类型: Kafka

* 地址: 172.16.101.137:9092,172.16.101.1

* 安全模式: 非认证模式

测试 保存 退出

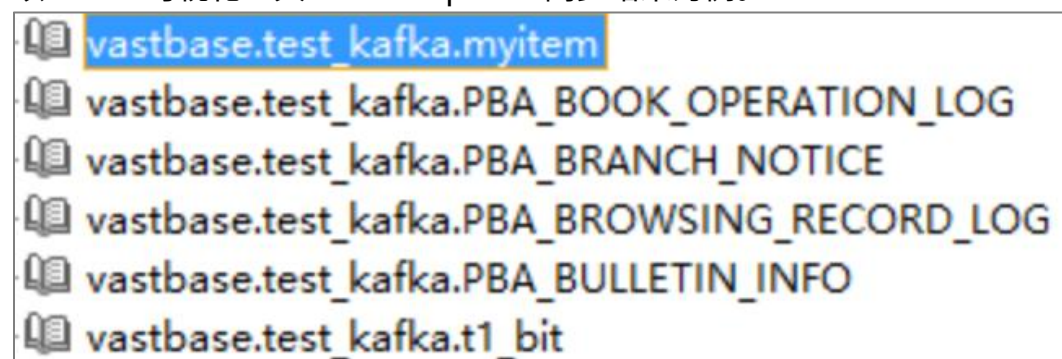
说明：集群环境地址 ip:9092 以逗号分割，数据源配置部分因支持集群的变动会影响之前的作业的同步，故之前的 kafka 数据源须重新创建。

- ❖ 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。

1.6.2.3.2.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 时空数组。
- newColumnValues: 列的新值。Key:Value 为字段名 column_name: 字段值 column_value,DELETE 时空数组。
- dataColumnType: 列的数据类型。Key:Value 为字段名 column_name: 字段数据类型 column_type。

1.6.2.4. 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。

数据库名	模式名	表名	分区数量
gbk	test_kafka	PBA_BROWSING_RECORD_LOG	1
gbk	test_kafka	PBA_BULLETIN_INFO	1
gbk	test_kafka	PBA_BULLETIN_INFO_BROWSE_RECORD	1
gbk	test_kafka	PBA_BULLETIN_INFO_LIKE_RECORD	1
gbk	test_kafka	PBA_BULLETIN_INFO_OPERATE_LOG	1
gbk	test_kafka	PBA_BULLETIN_INFO_VISIBLE_RANGE	1
gbk	test_kafka	test_monitor	2
gbk	test_kafka	t_normal2	4
gbk	test_kafka	t_normal3	6
gbk	test_kafka	t_normal_default1	8
gbk	test_kafka	t_normal_default2	10
gbk	test_kafka	t_normal_default3	1

共 15 条 < 1 > 15 条/页

提示: 单击分区数量进入编辑状态, 分区数量取值1~10并且只能增加不能减少

确定 退出

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”。

运行

操作: ☒ 增量迁移

运行 退出

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。

1.6.3. 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。

1.6.4. FAQ

1.6.4.1. Replication slot

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

报错信息：如图

```
[14:05:21:935] [ERROR] - cn.com.atlasdata.incremental.w.q.r(PgIncrementalCollectTask.java:70) - task:uaa_login_log#incremental#collect#incremental#1,任务运行出错,FATAL: all replication slots are in use
Hint: Free one or increase max replication slots.
cn.com.atlasdata.incremental.i.b: FATAL: all replication slots are in use
Hint: Free one or increase max replication slots.
at cn.com.atlasdata.incremental.i.b.A(CollectListener.java:78) ~[incremental.jar:?]
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) ~[?:1.8.0_322]
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) ~[?:1.8.0_322]
at java.lang.Thread.run(Thread.java:750) [?:1.8.0_322]
Caused by: org.postgresql.atlas.util.PSQLException: FATAL: all replication slots are in use
Hint: Free one or increase max replication slots.
at org.postgresql.atlas.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2795) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2534) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:315) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.jdbc.PgStatement.executeInternal(PgStatement.java:452) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.jdbc.PgStatement.execute(PgStatement.java:376) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.jdbc.PgStatement.executeWithFlags(PgStatement.java:318) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.jdbc.PgStatement.executeCachedSql(PgStatement.java:304) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.jdbc.PgStatement.executeWithFlags(PgStatement.java:281) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.jdbc.PgStatement.execute(PgStatement.java:277) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.replication.fluent.LogicalLogicalCreateSlotBuilder.make(LogicalCreateSlotBuilder.java:48) ~[gsjdb.jar:42.2.5]
at cn.com.atlasdata.incremental.a.a.eA(PgLogicalReplicaCollector.java:478) ~[incremental.jar:?]
at cn.com.atlasdata.incremental.a.a.cF(PgLogicalReplicaCollector.java:145) ~[incremental.jar:?]
at cn.com.atlasdata.incremental.i.b.A(CollectListener.java:72) ~[incremental.jar:?]
... 3 more
```

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

```
[14:44:02:959] [ERROR] - cn.com.atlasdata.incremental.a.a.ez(PgLogicalReplicaCollector.java:314) - task:test_kafka4#incremental#collect#incremental#1,
ERR00001,20220512a8ca,slot:slot_test_kafka4_incremental,recover-Exception:FATAL: replication slot "slot_test_kafka4_incremental" is already active
org.postgresql.atlas.util.PSQLException: FATAL: replication slot "slot_test_kafka4_incremental" is already active
at org.postgresql.atlas.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2795) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.core.v3.QueryExecutorImpl.processCopyResults(QueryExecutorImpl.java:1181) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.core.v3.QueryExecutorImpl.startCopy(QueryExecutorImpl.java:907) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.core.v3.replication.V3ReplicationProtocol.initializeReplication(V3ReplicationProtocol.java:58) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.core.v3.replication.V3ReplicationProtocol.startLogical(V3ReplicationProtocol.java:42) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.replication.fluent.ReplicationStreamBuilder$1.start(ReplicationStreamBuilder.java:38) ~[gsjdb.jar:42.2.5]
at org.postgresql.atlas.replication.fluent.LogicalLogicalStreamBuilder.start(LogicalStreamBuilder.java:37) ~[gsjdb.jar:42.2.5]
at cn.com.atlasdata.incremental.a.a.eC(PgLogicalReplicaCollector.java:534) ~[incremental.jar:7]
at cn.com.atlasdata.incremental.a.a.eC(PgLogicalReplicaCollector.java:308) ~[incremental.jar:7]
at cn.com.atlasdata.incremental.a.a.ez(PgLogicalReplicaCollector.java:224) ~[incremental.jar:7]
at cn.com.atlasdata.incremental.a.a.cF(PgLogicalReplicaCollector.java:166) ~[incremental.jar:7]
at cn.com.atlasdata.incremental.l.b.A(CollectListener.java:72) ~[incremental.jar:7]
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [?:1.8.0_322]
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [?:1.8.0_322]
at java.lang.Thread.run(Thread.java:750) [?:1.8.0_322]
```

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。

1.6.4.2. 数据库错误

❖ permission denied for table tb_kafka_partionconf

报错信息：如图

```
[15:57:30:188] [INFO] - cn.com.atlasdata.incremental.h.a.D(RpcHandler.java:159) - task:test_kafka1#incremental#apply#incremental#1,tasktype:apply,RPC请求参数:
{jobid=test_kafka1#incremental,tasktype=apply,thread=Y,jobtype=incremental,jobname=test_kafka1,taskid=test_kafka1#incremental#apply#incremental#1},应答参
数:{msg=task:test_kafka1#incremental#apply#incremental#1,执行任务发生错误:{otherMsg:"目标库错误",errorMsg:"ERROR: permission denied for table tb_kafka_partion
conf","logId":"2022051180b4"},errorCode=ERR00001,result=error,autoRestart=false,code=40002,errorcode=ERR00001,status=error}
```

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：

```
[root@server_101_9 upgrade]# su - appusr
上一次登录: 5月 23 09:58:36 CST 2022pts/12 上
[appusr@server_101_9 ~]$ cd /exbase/deployment/atlasdb/bin
[appusr@server_101_9 bin]$ ./psql -datlasdb -p31003
psql (11.5)
PSQL: Release 2.7.0
Connected to:
AtlasDB V2.7 Enterprise Edition Release - 64-bit Production
Type "help" for help.
```

2. 反向增量迁移前置步骤

2.1. MySQL 到 Vastbase G100 反向

2.1.1. 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

2.1.2. 操作步骤

2.1.2.1. 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。

2.1.2.2. 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。

```
host replication test 172.16.103.118/0 md5
"pg_hba.conf" 100L,4526c
```

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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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
```

2.1.2.3. 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;
```

2.2. MySQL 到 Vastbase E100 反向

2.2.1. 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

2.2.2. 操作步骤

2.2.2.1. 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。

2.2.2.2. 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
```

(2) 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。

```
[root@localhost protobuf-master]# find / -name protobuf.pc
/root/protobuf/protobuf-master/protobuf.pc
/usr/lib/pkgconfig/protobuf.pc

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_LIBRARY_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。

2.2.2.3. 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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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。

2.2.2.4. 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;

```

2.3. MySQL 到 PostgreSQL 反向

2.3.1. 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

2.3.2. 操作步骤

2.3.2.1. 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。

2.3.2.2. 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

(3) 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
```

(4) 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。

```
[root@localhost protobuf-master]# find / -name protobuf.pc
/root/protobuf/protobuf-master/protobuf.pc
/usr/lib/pkgconfig/protobuf.pc

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_LIBRARY_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。

2.3.2.3. 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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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。

2.3.2.4. 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;
```

2.4. Oracle 到 Vastbase G100 反向

2.4.1. 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

2.4.2. 操作步骤

2.4.2.1. 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。

2.4.2.2. 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。

```
host replication test 172.16.103.118/0 md5
"pg_hba.conf" 100L,4526c
```

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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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
```

2.4.2.3. 创建表

步骤 1 登录 Vastbase G100 目标库，登录用户需要与后续 exBase 平台使用的数据库连接用户保持一致。切换至连接目标库数据库的 database、数据库连接用户的 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库连接用户。

```
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 平台使用的数据库连接用户保持一致。切换至连接 Oracle 数据库的 database、数据库连接用户的 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
);
```

2.5. Oracle 到 Vastbase E100 反向

2.5.1. 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

2.5.2. 操作步骤

2.5.2.1. 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。

2.5.2.2. 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

(5) 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
```

(6) 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

```
[root@localhost protobuf-master]# find / -name protobuf.pc
/root/protobuf/protobuf-master/protobuf.pc
/usr/lib/pkgconfig/protobuf.pc
```

(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_LIBRARY_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。

2.5.2.3. 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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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。

2.5.2.4. 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
);

```

2.6. Oracle 到 PostgreSQL 反向

2.6.1. 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
----	-----

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

2.6.2. 操作步骤

2.6.2.1. 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。

2.6.2.2. 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

(7) 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
```

(8) 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

```
[root@localhost protobuf-master]# find / -name protobuf.pc
/root/protobuf/protobuf-master/protobuf.pc
/usr/lib/pkgconfig/protobuf.pc
```

(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_LIBRARY_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。

2.6.2.3. 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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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。

2.6.2.4. 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,
updatetip timestamp
);

```

2.7. SQL Server 到 Vastbase G100 反向

2.7.1. 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

2.7.2. 操作步骤

2.7.2.1. 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 到 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。

2.7.2.2. 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。

```
host replication test 172.16.103.110/0 md5
"pg_hba.conf" 100L,4526c
```

步骤 2 修改数据库配置

目标库 Vastbase G100 在 postgresql.conf 配置文件中相关配置。

- ❖ 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。

```
#max_wal_senders=4
```

```
#max_replication_slots=10
```

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