openGauss 发布订阅支持基础数据同步特性设计说明书

所属SIG组:	storage
落入版本:	3.0.0
设计人员:	陈晓滨
日期:	2022.4.25

改版记录

日期	修订版本	修订描述	作者	审核
2022.4.25	1.0.0	初稿	陈晓滨	SIG组成员

目录

3.1目标

3.2总体方案

4.Use Case一实现

4.1设计思路

4.2约束条件

4.3详细实现(从用户入口的模块级别或进程级别消息序列图)

4.4子系统间接口(主要覆盖模块接口定义)

4.5子系统详细设计

4.6DFX属性设计

4.6.1性能设计

4.6.2升级与扩容设计

4.6.3异常处理设计

4.6.4资源管理相关设计

4.6.5小型化设计

4.6.6可测性设计

4.6.7安全设计

4.7系统外部接口

4.8自测用例设计

5.Use Case二实现

6.可靠性&可用性设计

6.1冗余设计

6.2故障管理

6.3过载控制设计

6.4升级不中断业务

6.5人因差错设计

6.6故障预测预防设计

7.安全&隐私&韧性设计

7.1Low Level威胁分析及设计

7.1.12层数据流图

7.1.2业务场景及信任边界说明

7.1.3外部交互方分析

7.1.4数据流分析

7.1.5处理过程分析

7.1.6数据存储分析

表目录

图目录

List of abbreviations 缩略语清单：

Abbreviations 缩略语	Full spelling 英文全名	Chinese explanation 中文解释

1.特性概述

1.1范围

openGauss 发布订阅支持基础数据同步。

1.2特性需求列表

表1：特性需求列表

需求编号	需求名称	特性描述	备注
1	发布订阅支持基础数据同步	创建订阅时通过copy的方式复制表的初始数据	

2.需求场景分析

2.1特性需求来源与价值概述

当前openGauss发布订阅还不支持基础数据复制。

2.2特性场景分析

【关键场景】openGauss 两个数据库分别创建发布订阅，发布表的初始数据复制到订阅端。

2.3特性影响分析

与其他需求及特性的交互分析如下：

平台差异性分析：不涉及

兼容性分析：兼容以前的版本

约束及限制：除支持同步基础数据外，继承发布订阅的所有约束

2.3.1硬件限制

不涉及

2.3.2技术限制

不涉及

2.3.3对License的影响分析

不涉及

2.3.4对系统性能规格的影响分析

创建订阅时部分系统资源会用于基础数据复制，复制期间对性能有所影响。

2.3.5对系统可靠性规格的影响分析

不涉及

2.3.6对系统兼容性的影响分析

不涉及

2.3.7与其他重大特性的交互性，冲突性的影响分析

WalSender流程有所变化，除了流复制走WalSndLoop外，执行常规的PostgreMain，以此处理Q报文的消息，执行SQL。

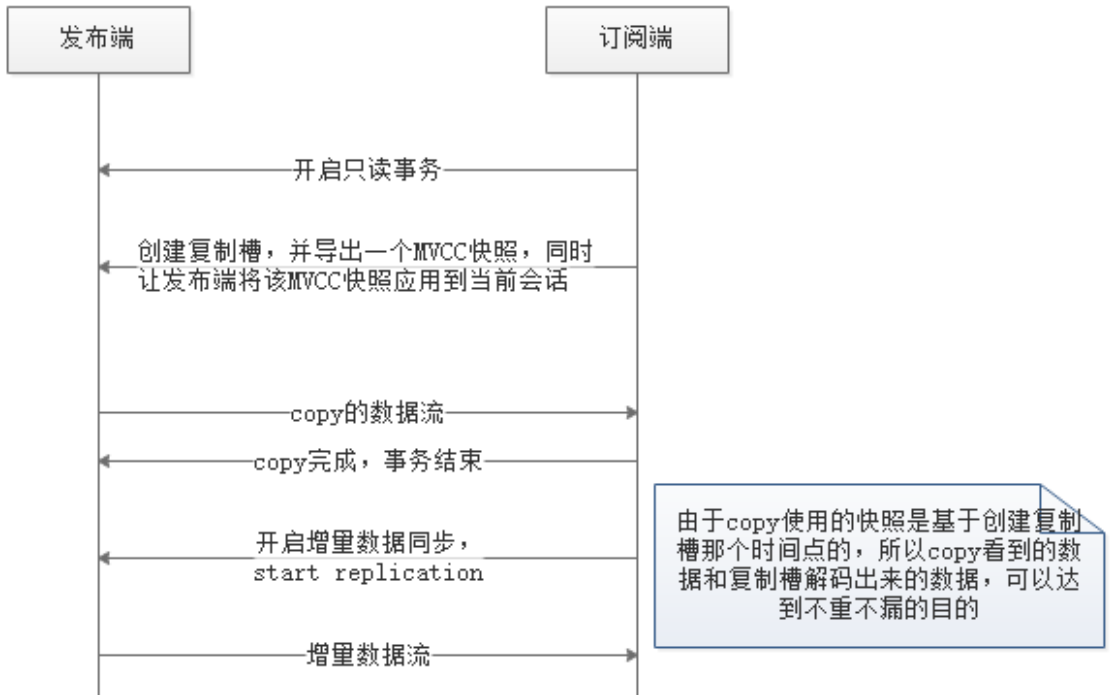
2.4同类社区/商用软件实现方案分析

PostgreSQL原生的基础数据复制功能与本方案类似，不同之处在于快照的生成和使用。

当前逻辑复制有一套完整的快照生成逻辑，用于判断系统表的可见性（快照类型为SNAPSHOT_HISTORIC_MVCC），主要是记录快照的xmin、xmax、xip。其中xip是在[xmin,xmax)之间的已提交的事务号数组。这一点PG和openGauss是一样的。

关于如何找到一致性点，使得基础数据和后续的增量数据不重不漏，当前PG是利用已有的HISTORIC快照信息，组成一个普通的MVCC快照，用于普通数据的可见性判断。原理是HISTORIC快照中记录的是[xmin,xmax)间已提交的事务号数组，而普通MVCC快照中记录的是[xmin,xmax)间活跃的事务号数组，那么只需要利用HISTORIC快照的xmin、xmax、xip信息，遍历xmin到xmax，将不在xip中的事务当成活跃事务即可。这样就简单的利用HISTORIC快照生成了一个MVCC快照，此快照可以用于后续的数据可见性判断。

PG的快照使用逻辑为：



由于openGauss对于MVCC可见性判断没有使用活跃事务链表的方式，而是CSN的方式，所以在组成MVCC快照过程中不再反转xip数组，获取活跃事务，而是获取xip数组中每个已提交事务的csn，取其中最大csn为快照的snapshotcsn，具体实现在下文展开。

3.特性/功能实现原理(可分解出来多个Use Case)

3.1目标

openGauss支持基础数据复制。

3.2总体方案

基础数据复制的主要问题在于如果找到一个一致性点，使得基础数据和后续的增量数据不重不漏，这里参考《GaussDB Kernel V500R001C00 逻辑复制特性设计说明书.docx》和PostgreSQL的方案，使用导出快照及csn的方式，增量复制从此位置之后完成衔接。

4.Use Case一实现

4.1设计思路

当前订阅端的apply worker负责增量数据同步，由于基础数据复制和增量数据复制的逻辑存在部分重叠，可以将基础数据的复制功能合并到apply worker。

目前负责同步增量数据的apply worker是针对一个订阅中的所有表，即一个订阅中的所有表的增量数据都由apply worker来进行处理，而基础数据复制时，由于数据量大，而且是通过copy的方式进行，故设计成一个表一个worker，并增加基础数据复制的并行度。LogicalRepWorker结构体中的relid，表示正在被同步的表的oid，当oid为invalid时，表明是处理所有表增量数据复制的apply worker，否则是处理单个表全量数据复制的sycn worker。

当创建一个订阅时，新增选项copy_data，表明是否需要同步基础数据，默认为true。新增系统表pg_subscription_rel（非共享），用于记录所有被订阅表的状态信息。

4.2约束条件

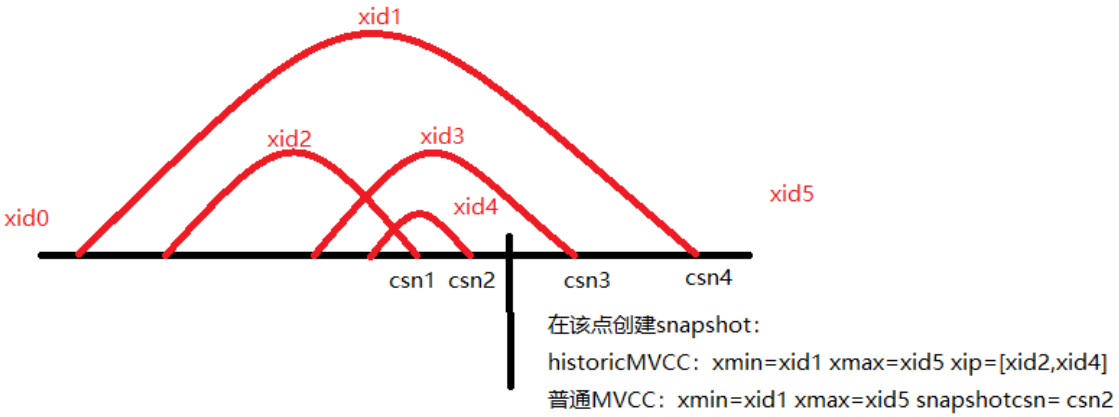
除支持同步基础数据外，继承发布订阅的所有约束

4.3详细实现(从用户入口的模块级别或进程级别消息序列图)

4.3.1 全量复制快照确定

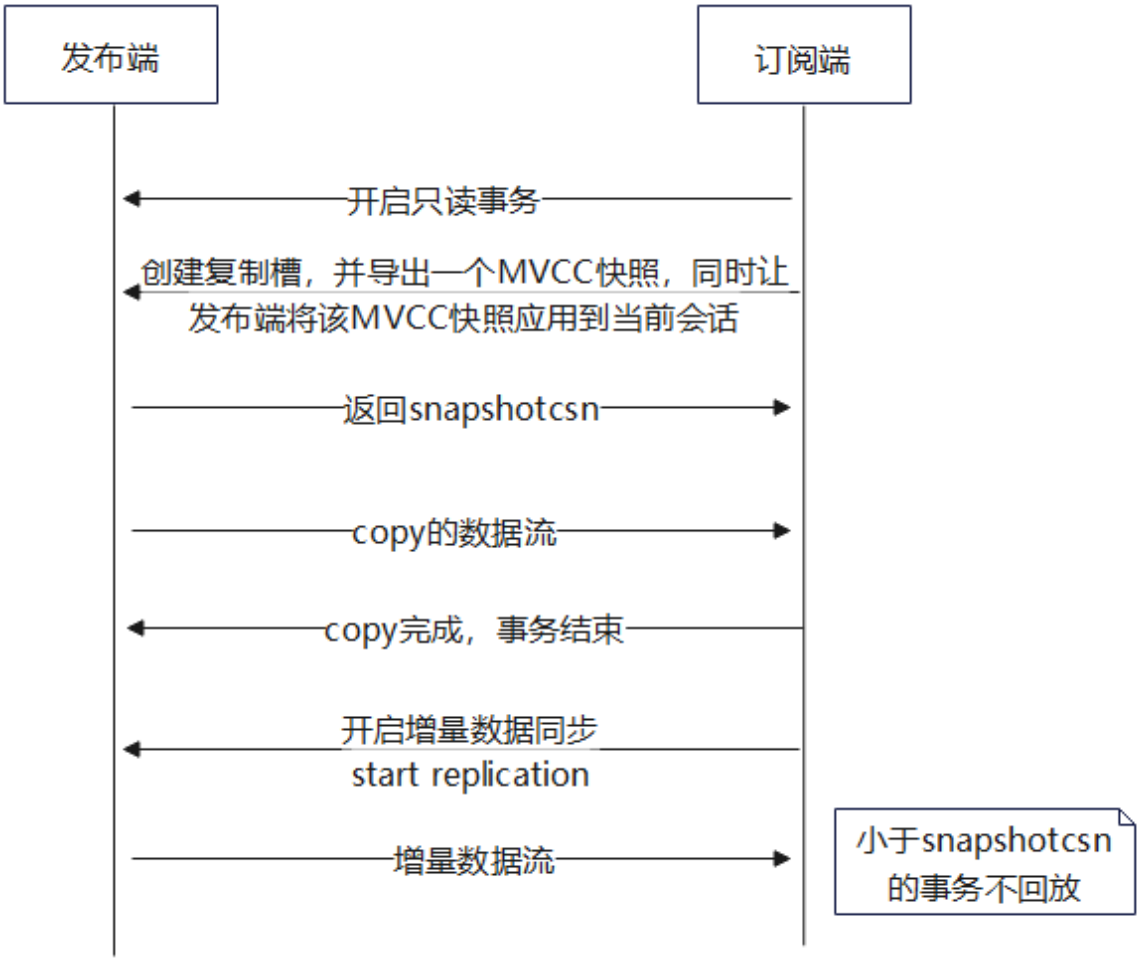
保证基础数据复制和后续的增量数据不重不漏，是该特性需要解决的一个重要问题。全量复制是通过在copy事务中应用快照的方式，复制该快照可见的所有数据；增量复制是从复制槽的restart_lsn开始读取日志，并且只解码confirmed_flush之后提交的事务。因此全量+增量数据不重不漏的关键在于保证confirm_flush之前提交的所有事务都在全量复制的快照中可见，confirmed_flush之后提交的事务不会重复解码。

因此对比PostgreSQL的方案（见2.4），openGauss的设计如下：同样在创建复制槽时会构建一个HISTORIC快照，其中xip数组记录了[xmin,xmax)间已提交的事务号， 利用该快照组成普通MVCC快照的过程中，snapshotcsn取HISTORIC快照的xip数组中最大的csn，判断可见性时，当csn小于snapshotcsn时，则该事务可见，反之不可见。如上图，全量复制利用该MVCC快照，可以同步csn1、csn2的事务。



当前openGauss引入了csn，由于csn乱序落盘导致无法依赖本地xlog文件计算出精确的snapshot。

上图是模拟事务在xlog的落盘顺序，每一个块表示一个提交的事务，上面记录的是其csn，csn1 < csn2 < csn3 < csn4。假设仍在csn2处创建快照，将snapshotcsn置为csn2，那么全量复制会解出csn1和csn2，之后增量复制会解出csn1、csn3和csn4，可见csn1被重复解码。因此，在创建好快照之后，增量复制需要过滤掉小于snapshotcsn的所有事务。综上，全量+增量的流程图如下。



4.3.2 新增系统表pg_subscription_rel

pg_subscription_rel结构如下：

列名	类型	描述
srsubid	oid	订阅的oid
srrelid	oid	表的oid
srsubstate	Char	表的同步状态。'i' 表示initializing，待复制基础数据；'d'表示正在同步基础数据（data is being synchronized）；'f'表示基础数据copy完成；'s'表示该表同步完成（同步位置超过apply worker）；'r'表示可以开始增量数据复制
csn	CommitSeqNo	同步基础数据时使用的快照的csn信息
srsublsn	XLogRecPtr	发布端的lsn，用于当同步状态在's'或者'r'时，判断是否进行状态切换，其他场景为NULL

4.3.3 创建订阅流程变更

1）解析命令获取对应参数

2）权限校验，创建订阅必须为超级用户

3）校验订阅名称，连接信息

4）向系统表pg_subscription插入订阅信息

5）创建复制源

6）连接发布端

7*）向发布端发送sql，查询发布端发布的所有表的表名、schema名

8*）对于每一个表，获取表的oid信息，检查表能否被订阅（如检查是否为普通表等）

9*）检查完后，将表的信息插入到pg_subscription_rel中，对于srsubstate，如果copy_data为true，记录状态为'i'，表示待复制基础数据，否则记录状态为'r'，表示可以开始增量数据复制

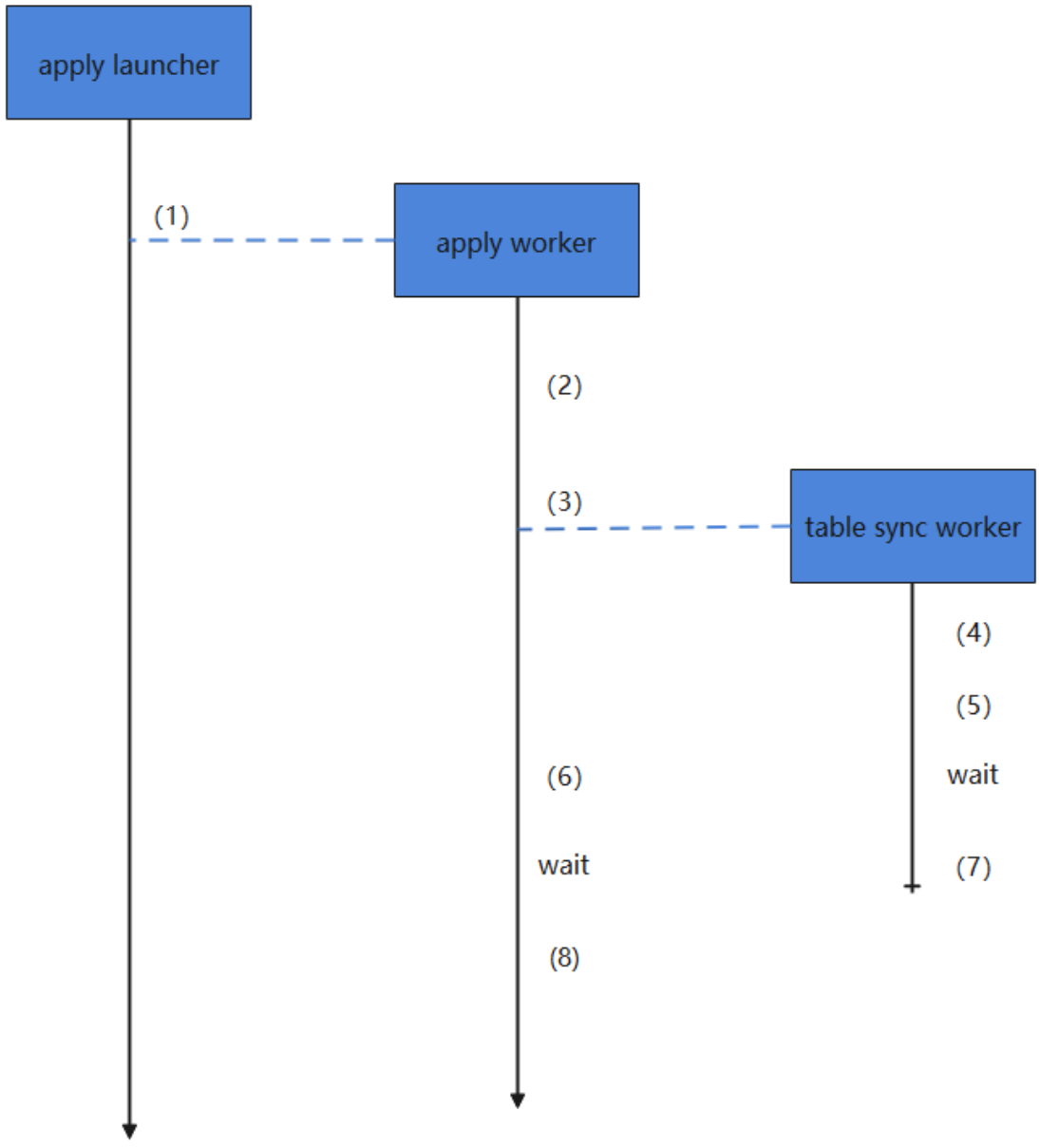
10）创建复制槽

11）提交时唤醒launcher线程

主要是新增流程7~9，其余流程和原始流程一致。

4.3.4 基础数据复制流程

launcher线程的主要流程不变。为方便理解，当apply worker承担同步基础数据时，称其为 sync worker，当apply worker承担同步增量数据时，称其为 apply worker。同步基础数据的流程如下。



1）launcher线程遍历pg_subscription系统表，对于每个订阅，若没有apply线程则创建新的apply线程并启动，赋予LogicalRepWorker中的relid为invalid，表明是一个同步增量数据的apply worker

2）apply worker扫描pg_subscription_rel表，获取所有同步状态不是'r'的表，并为其启动一个sync worker，赋予LogicalRepWorker中的relid为对应表的oid，表明是一个同步基础数据的sync worker。apply worker可继续处理其他已经完成基础数据同步表的增量数据。

> 注意sync worker的数量需要有所限制，新增guc参数max_sync_workers_per_subscription，控制每个订阅同时存在的最大的worker数，避免同步基础数据时起了太多线程，占用过多资源。

3）启动table sync worker，创建逻辑复制槽和复制源（replication origin，参考《GaussDB Kernel V500R002C10 openGauss开源回合设计说明书.docx》的3.1.3.5 跟踪复制进度一节），将表的订阅状态由INIT状态更新为DATASYNC同步

4）sync线程创建临时复制槽并获取起始LSN位置和snapshotcsn，同步基础数据，命令为COPY (SELECT * FROM table_name) TO STDOUT

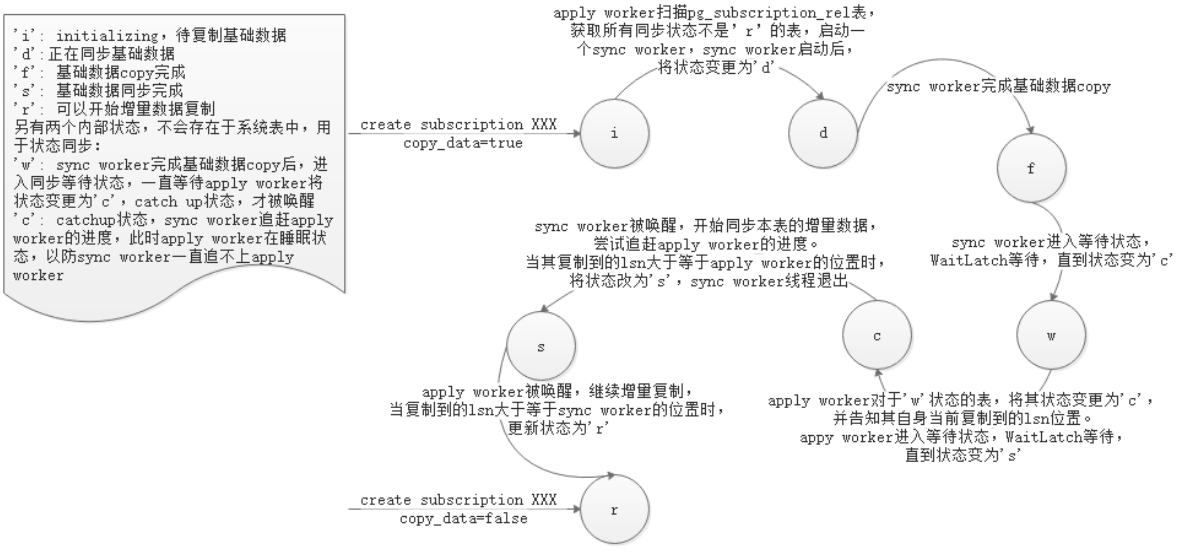
5）sync线程基础数据初步同步完成，更新sync线程状态为SYNCWAIT即同步等待状态，sync线程进入休眠状态，等待apply线程更新状态为SUBREL_STATE_CATCHUP

6）apply线程遍历pg_subscription_rel，获取所有没有同步完基础数据的表信息，发现处于SYNC_WAIT状态的sync线程，更新其srsublsn为当前apply线程的最新位置，判断首次同步到的基础数据的位置如果落后于当前的最新位置，更新状态为CATCHUP，并唤醒sync线程。apply线程进入休眠状态，等待sync线程更新状态为SYNC_DONE。

7）sync线程从首次同步到的位置开始追赶apply线程的最新LSN位置，并过滤掉小于步骤4获取的snapshotcsn的事务，完成追赶后记录最新的LSN位置，修改状态为SYNC_DONE，唤醒apply线程，然后退出。

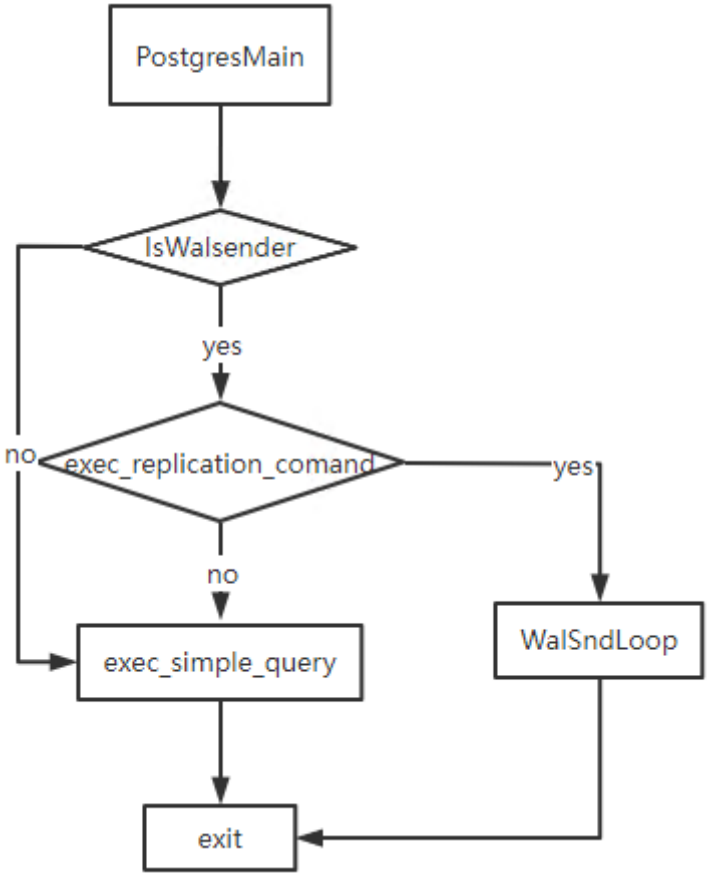
8）apply线程更新状态为READY，开始持续的WAL同步

订阅表的状态变迁逻辑如下：



4.3.5 WalSender改造

上述很多实现是基于walsender具备执行SQL语句的能力，4.3.1中开启事务和事务提交，以及4.3.3向发布端发送sql，查询发布端发布的所有表的表名、schema名。参考PostgreSQL的commit：fd5942c18f977a36fec66a8d1281092805f2a55e，使WalSender也走常规的PostgreMain流程，基本流程如下。



之前在PostgresMain中判断如果是walsender，直接进WalSndLoop。当前改动是会先判断是否是流复制相关的命令，如果是，依然走WalSndLoop；如果不是，且消息是Q报文，则与普通worker一样，调exec_simple_query执行。仅支持walsender处理Q、H、S、X、d、c、f、EOF报文的消息，其他如P、B、E、F、C、D均不支持。

该改动除了支撑发布订阅基础数据复制过程中的SQL执行，还有一点好处是相比WalSndLoop，PostgreMain能更好地处理信号和错误恢复，例如，如果客户端挂起并且在开始流传输之前收到SIGTERM，则walsender可立即终止，而不是挂起直到连接超时。

4.3.6 subscription新增语法

4.3.6.1 create subscription

新增选项copy_data和connect，说明如下：

```
copy_data (boolean)

    指定在复制启动后是否应复制正在订阅的发布中的现有数据。 默认值是`true`。

connect  (boolean)

    指定CREATE SUBSCRIPTION是否应该连接到发布者。 将其设置为false将会改变默认值enabled和copy_data为false。

    不允许将connect设置为false的同时将enabled或copy_data 设置为true。

    因为该选项设置为false时不会建立连接，因此表没有被订阅，所以当启用订阅后，不会复制任何内容。需要运行ALTER SUBSCRIPTION ... REFRESH PUBLICATION才能订阅表。
```

4.3.6.2 alter subscription

新增语法refresh publication，说明如下：

```
refresh publication

    从发布者获取缺少的表信息。这将开始复制自上次调用REFRESH PUBLICATION 或从CREATE SUBSCRIPTION以来添加到订阅发布中的表。

    refresh_option指定了刷新操作的附加选项。 支持的选项有：

    copy_data (boolean)

        指定在复制启动后是否应复制正在订阅的发布中的现有数据。 默认值是true。（以前订阅的表不会被复制。）
```

fresh流程与4.3.3和4.3.4描述的一样，当执行fresh命令时，会向发布端发送SQL，查询发布端新发布的表，将新表添加到pg_subscription_rel中，再由apply worker遍历pg_subscription_rel，找到新增的未同步的表，开始复制。

4.4子系统间接口(主要覆盖模块接口定义)

不涉及

4.5子系统详细设计

不涉及

4.6DFX属性设计

4.6.1性能设计

待后续实现摸底

4.6.2升级与扩容设计

新增了系统表pg_subscription_rel，因此需要适配升级。添加前置升级脚本，创建系统表；添加前置回滚脚本，删除该系统表。

4.6.3异常处理设计

1. 订阅端在copy阶段遇到故障时，copy事务会中断，重启后会重建复制槽和复制源，并执行完整的copy流程
2. sync worker在追赶apply worker阶段遇到故障，重启后由于复制源保存着之前同步的进度，因此可继续追赶。

4.6.4资源管理相关设计

不涉及

4.6.5小型化设计

不支持小型化

4.6.6可测性设计

见4.8

4.6.7安全设计

不涉及

4.7系统外部接口

1. 新增guc参数max_sync_workers_per_subscription，控制每个订阅同时存在的最大的worker数。
2. 见4.3.6，subscription新增语法

4.8自测用例设计

1. 发布端创建若干表，并插入数据。订阅端同样创建表
2. 创建发布订阅，并将copy_data选项打开，订阅端查询，观察基础数据被同步。
3. 发布端插入数据，订阅端查询，观察表数据无重复、遗漏问题。
4. 发布端新增表，并插入数据，订阅端同样创建该表
5. 执行fresh命令，并将copy_data选项打开，订阅端查询，观察新增的表的基础数据被同步。
6. 发布端插入数据，订阅端查询，观察表数据无重复、遗漏问题，发布订阅正常。

5.Use Case二实现

*同第**4** 章*

6.可靠性&可用性设计

6.1冗余设计

不涉及

6.2故障管理

不涉及

6.3过载控制设计

guc参数max_sync_workers_per_subscription，控制每个订阅同时存在的最大的worker数，避免同步基础数据时起了太多线程，占用过多资源。

6.4升级不中断业务

不涉及

6.5人因差错设计

不涉及

6.6故障预测预防设计

不涉及

7.安全&隐私&韧性设计

不涉及的需要说明原因或者简要说明存在的风险(通常低风险且无对应消减建议的可不写当前章节)

7.1Low Level威胁分析及设计

7.1.12层数据流图

7.1.2业务场景及信任边界说明

不涉及

7.1.3外部交互方分析

不涉及

7.1.4数据流分析

不涉及

7.1.5处理过程分析

不涉及

7.1.6数据存储分析

不涉及

7.1.7缺陷列表

不涉及

7.2隐私风险分析与设计

7.2.1隐私风险预分析问卷

序号	问题	是否满足	填写指导
1	该产品是否收集或处理个人数据	否	
2	上一版本是否做过隐私风险分析	是	
3	当前版本是否有新增特性收集或处理个人数据	否	
4	当前版本是否存在个人数据收集范围发生变化	否	

7.2.2隐私风险预分析总结

当前版本不涉及个人数据收集，因此该版本无需做隐私风险分析。

7.2.3个人数据列表

不涉及

7.2.4XX需求设计

7.2.4.1需求说明

不涉及

7.2.4.2需求设计

不涉及

7.2.5YY需求设计

8.特性非功能性质量属性相关设计

8.1可测试性

参考4.8自测用例设计

8.2可服务性

开发者指南中发布订阅章节会对新增特性做说明

8.3可演进性

不涉及

8.4开放性

不涉及

8.5兼容性

前向兼容

8.6可伸缩性/可扩展性

不涉及

8.7可维护性

不涉及

8.8资料

类别	手册名称	是否涉及 (Y/N)	具体修改或新增内容简述
白皮书	技术白皮书	N	
产品文档	产品描述	N	
	特性描述	N	
	编译指导书	N	
	安装指南	N	
	管理员指南	N	
	开发者指南 （包括开发教程、SQL参考、系统表和系统视图、GUC参数说明、错误码说明、API参考等）	Y	subscription章节添加语法说明、新增guc参数
	工具参考	N	
	术语表	N	
入门	简易教程	N	

9.数据结构设计（可选）

不涉及

10.参考资料清单