第九节：组复制要求和限制---基础篇

1.组复制要求：

要用于组复制的实例必须满足以下要求：

基础配置：

InnoDB存储引擎：

数据必须存储在InnoDB事务存储引擎中。采用乐观锁的方式来执行事务，然后在事务提交时检查是否存在冲突。如果存在冲突，为了保持整个集群的数据一致性，将回滚一些事务。（存在冲突的事务中，先提交的事务不会受到影响，继续完成提交，而后提交的事务会被回滚）这需要事务存储引擎。此外，InnoDB提供了一些附加功能，可以在与组复制一起操作时更好地管理和处理冲突的事务。使用其他存储引擎（包括临时的MEMORY存储引擎），由于缺乏对某些事务层面特性的支持，可能会导致组复制中的错误。可以通过在组成员上设置disabled_storage_engines系统变量来禁用其他存储引擎，例如：

```
disabled_storage_engines="MyISAM,BLACKHOLE,FEDERATED,ARCHIVE,MEMORY"
```

主键:

该集群要复制的每个表都必须具有定义的主键，或等效的主键，其中等效主键是非null的唯一键。此类键是作为表中每一行数据的唯一标识符，从而使系统能够通过该唯一标识符准确标识每个事务已修改的行来确定哪些事务发生冲突。组复制具有自己的内置的主键或主键等效项检查集，并且不使用sql_require_primary_key系统变量执行的检查。

可以将运行组复制实例的sql_require_primary_key = ON设置为ON，并且可以将组复制通道的CHANGE MASTER TO语句的REQUIRE_TABLE_PRIMARY_KEY_CHECK选项设置为ON。但当设置sql_require_primary_key = ON或REQUIRE_TABLE_PRIMARY_KEY_CHECK = ON时，在执行的检查下可能会不允许某些组复制的内置检查所允许的事务。

网络性能：

MySQL组复制要求部署在实例彼此非常接近的集群环境中。集群的性能和稳定性可能会受到网络延迟和网络带宽的影响。所有组成员之间必须始终保持双向通信。如果针对实例阻止了入站或出站通信（例如，通过防火墙或由于连接问题），则该节点无法在该集群中运行，并且该组成员（包括有问题的节点）可能无法报告受影响实例的正确状态。

从MySQL 8.0.14开始，可以使用IPv4或IPv6网络基础结构或两者的结合来进行远程组复制服务器之间的TCP通信。

同样从MySQL 8.0.14（组复制实例位于同一位置并共享本地组通信引擎（XCom）实例）开始，在可能的情况下，使用开销较小的专用输入通道代替TCP套接字进行通信。对于某些需要在远程XCom实例之间进行通信的组复制任务（例如加入组），仍然使用TCP网络，因此网络性能会影响集群的性能。

MySQL server配置：

必须按照作为组成员的服实例上所示配置以下变量：

server唯一标识符：根据复制拓扑中所有实例的要求，使用server_id系统变量为实例配置唯一server ID。server ID必须是介于1和（232）-1之间的正整数，并且必须与复制拓扑中的任何其他实例使用的每个其他server ID不同。

二进制日志处于开启状态：设置--log-bin [= log_file_name]。 MySQL Group Replication复制二进制日志内容，因此二进制日志需要打开才能运行。默认情况下启用此选项。

备库的所有更新都要记录到其binlog中：设置--log-slave-updates。默认情况下启用此选项。组成员需要记录加入时从其引导节点收到并通过复制应用程序应用的事务，记录他们从集群中接收并应用的所有事务。这样，组复制就可以通过从现有组成员的二进制日志进行状态转移来执行分布式恢复。

二进制日志行格式：设置--binlog-format = row。组复制依靠基于行的复制格式在集群中的实例之间一致地传播更改。它依靠基于行的基础结构来提取必要的信息，以检测在集群中的不同实例中同时执行的事务之间的冲突。从MySQL 8.0.19起，将REQUIRE_ROW_FORMAT设置自动添加到组复制的通道中，以在应用事务时强制使用基于行的复制。

关闭二进制日志checksum校验：（适用于MySQL 8.0.20）。直到MySQL 8.0.20，设置--binlog-checksum = NONE。在这些发行版中，组复制无法使用校验和，并且不支持二进制日志中的校验和。从MySQL 8.0.21开始，组复制支持校验和，因此组成员可以使用默认设置。

打开gtid复制：设置gtid_mode = ON，然后设置force_gtid_consistency = ON。组复制使用全局事务标识符来准确跟踪每个实例上已提交哪些事务，从而能够推断出哪些服务器已执行可能与其他位置已提交的事务发生冲突的事务。

复制信息存储到表中：设置master_info_repository = TABLE和relay_log_info_repository = TABLE。复制应用程序需要将复制元数据写入mysql.slave_master_info和mysql.slave_relay_log_info系统表。这样可以确保组复制插件对复制元数据具有一致的可恢复性和事务管理。从MySQL 8.0.2开始，这些选项默认设置为TABLE，而从MySQL 8.0.3开始，不推荐使用FILE设置。

事务写集提取：设置--transaction-write-set-extraction = XXHASH64，以便在收集行以将其记录到二进制日志时，服务器也将收集写集。写集基于每行的主键，并且是标签的简化且紧凑的视图，该标签唯一地标识已更改的行。然后，该标签用于检测冲突。默认情况下启用此选项。

二进制日志依赖跟踪：设置binlog_transaction_dependency_tracking = WRITESET_SESSION可以提高组成员的性能，具体取决于集群的负载。在应用中继日志中的事务时，组复制会在认证后执行自己的并行化，而与binlog_transaction_dependency_tracking设置的值无关。但是，binlog_transaction_dependency_tracking的值的确会影响如何将事务写入组复制节点上的二进制日志。这些日志中的依赖项信息用于协助从引导节点的二进制日志进行状态转移的过程，以进行分布式恢复，只要节点加入或重新加入该集群，就会发生这种情况。

默认表加密：在所有组成员上将--default-table-encryption设置为相同的值。只要所有节点上的设置都相同，就可以启用（ON）或禁用（默认，OFF）默认模式和表空间加密。

设置表名称小写：在所有组成员上将--lower-case-table-names设置为相同的值。设置1对于使用InnoDB存储引擎是正确的，这对于组复制是必需的。请注意，该设置并非在所有平台上都是默认设置。

多线程回放：可以将组复制节点配置为多线程回放，从而使事务可以并行应用。slave_parallel_workers的非零值将在节点上启用多线程回放，并且最多可以指定1024个并行应用线程。设置slave_preserve_commit_order = 1可以确保并行事务的最终提交与原始事务的顺序相同，这是组复制所必需的，它依赖于围绕所有参与节点以相同顺序接收和应用已提交事务的保证的一致性机制。最后，slave_preserve_commit_order = 1需要设置slave_parallel_type = LOGICAL_CLOCK，该设置指定用于决定允许哪些事务在只读节点上并行执行的策略。设置slave_parallel_workers = 0将禁用并行回放，并为副本提供一个单一的应用程序线程，而没有协调程序线程。使用该设置，slave_parallel_type和slave_preserve_commit_order选项无效，将被忽略。

PS：通过前面的章节我们可以知道，组复制是基于主从复制的基础架构实现的，但相对于主从复制的IO和SQL线程来讲，组复制中对原来接收主库二进制日志的IO线程改动较大（用了一些后台线程相互协作来代替了主从复制拓扑中的IO线程的功能），对解析二进制日志并进行回放的SQL线程（或者说协调器线程与worker线程）改动较小。每一个组成员中都会创建到组复制的连接（主从复制拓扑中的IO线程被替换为了一些接收、验证写集的一些后台线程，可通过performance_schema.threads表进行查看），每一个组成员对接收到的写集进行冲突认证检测，并将认证通过的写集（二进制日志）写入自身的中继日志中，然后，由SQL线程读取中继日志进行回放（多线程复制中，由协调器线程读取中继日志，然后并行分发给worker线程进行回放）。

2.组复制限制：

组复制存在以下已知限制：

在故障转移期间，针对多主模式组描述的限制和问题也可以适用于单主模式集群，尤其是在新选举的primary节点会从旧的primary节点中清除其applier队列时，在这个临界点，可以将其类比组复制中有2个成员可写。

--upgrade=minimal：当MySQL Server指定--upgrade=minimal选项启动时，如果发现需要执行更新，则，在执行升级操作完成之后，可能会导致组复制无法启动，因为minimal选项在执行更新时，只会更新数据字典、information_schema、performance_schema，但不会更新组复制内部所依赖的系统表（--upgrade选项在MySQL 8.0.16版本引入，之后，升级操作将不再需要单独使用mysql_upgrade工具，默认情况下--upgrade选项值为AUTO，表示自动判断是否需要执行完整的更新操作）。

间隙锁。组复制的并发事务认证过程未考虑间隙锁，因为有关间隙锁的信息在InnoDB之外不可用。

ps：对于处于多主模式的集群，除非在应用程序中依赖REPEATABLE READ语义，否则建议对组复制使用READ COMMITTED隔离级别。 InnoDB不使用READ COMMITTED中的间隙锁，它将InnoDB中的本地冲突检测与组复制执行的分布式冲突检测保持一致。对于单主模式的集群，只有primary节点接受写操作，因此READ COMMITTED隔离级别对组复制并不重要。

表锁和命名锁：冲突认证检测过程不考虑表锁（表锁指的是：使用lock...table语句加锁、使用unlock table语句解锁）或命名锁(命名锁指的是：使用GET_LOCK()函数创建、使用RELEASE_LOCK()函数释放的锁)。

复制中event的checksum校验：直到MySQL 8.0.20，组复制无法使用校验和，并且不支持二进制日志中的校验和，因此在将实例配置为组成员时必须设置binlog_checksum = NONE。从MySQL 8.0.21开始，组复制支持校验和，因此组成员可以使用默认设置binlog_checksum = CRC32。对于集群的所有节点，binlog_checksum的设置可以不同。

当校验可用时，组复制将不使用它们来验证group_replication_applier通道上的传入事件，因为事件是从多个源写入中继日志的，并且在它们在实际写入原始服务器的二进制日志之前就已写入。校验用于验证group_replication_recovery通道和组成员上任何其他复制通道上事件的完整性。

串行化隔离级别：默认情况下，多主集群不支持SERIALIZABLE隔离级别。将事务隔离级别设置为SERIALIZABLE将组复制将拒绝提交该事务。

并行DDL与DML操作：使用多主模式时，不支持针对同一对象但在不同实例上执行的并发数据定义语句和数据操作语句。在对象上执行数据定义语言（DDL）语句期间，在同一对象上但在不同服务器实例上执行并发数据操作语言（DML）可能会导致未检测到在不同实例上执行的DDL冲突的风险。

- 注意：在同一个组成员中对同一个对象并行执行DDL和DML语句，会由本地Server自行通过锁进行管理，不需要集群参与。

具有级联约束的外键：多主模式集群（所有成员均配置有group_replication_single_primary_mode = OFF）不支持具有多级外键依赖关系的表，特别是定义了CASCADING外键约束的表。这是因为导致由多主模式集群执行级联操作的外键约束可能导致无法检测到的冲突，并导致该组成员之间的数据不一致。因此，建议在用于多主模式集群的实例上将group_replication_enforce_update_everywhere_checks = ON设置为ON，以避免未检测到的冲突。

在单主模式下，这不是问题，因为它不允许同时写入集群中的多个节点，因此不存在未检测到冲突的风险。

多主模式死锁：当集群以多主模式运行时，SELECT .. FOR UPDATE语句可能导致死锁。这是因为该锁未在集群上不同的节点之间实现共享。

复制过滤：不能在组复制的成员中对组复制专用的group_replication_applier或group_replication_recovery通道配置使用全局复制过滤器，因为在某些组成员上过滤了事务会破坏组中所有成员之间的数据一致性。但是，如果某个组成员同时作为主从复制拓扑中的从库时，则该主从复制通道允许配置使用全局复制过滤器（这里的主从复制通道，不包括组复制专用的group_replication_applier或group_replication_recovery通道）。

加密连接：从MySQL 8.0.16开始，MySQL Server中提供了对TLSv1.3协议的支持，前提是MySQL是使用OpenSSL 1.1.1或更高版本进行编译的。在MySQL 8.0.16和MySQL 8.0.17中，如果服务器支持TLSv1.3，则组通信引擎中不支持该协议，则组复制不能使用该协议。组复制支持MySQL 8.0.18中的TLSv1.3，可将其用于组通信连接和分布式恢复连接。

在MySQL 8.0.18中，TLSv1.3可以在组复制中用于分布式恢复连接，但是group_replication_recovery_tls_version和group_replication_recovery_tls_ciphersuites系统变量不可用。因此，引导节点所在服务器必须允许使用至少一个默认启用的TLSv1.3密码套件。从MySQL 8.0.19开始，可以使用这些选项来配置对任何密码套件选择的客户端支持，如果需要的话，仅包括非默认密码套件。

如果在组复制中使用了TLSv1.3版本协议进行分布式恢复，则组复制的组成员中至少一个节点必须允许使用TLSv1.3版本协议的密码套件。

克隆操作：组复制启动并管理用于分布式恢复的克隆操作，但是已设置为支持克隆的组成员也可能参与用户手动启动的克隆操作。在MySQL 8.0.20之前的版本中，如果操作涉及正在运行“组复制”的组成员，则无法手动启动克隆操作。从MySQL 8.0.20开始，只要克隆操作不会删除和替换接收者上的数据，就可以执行此操作。因此，如果正在运行组复制，则用于启动克隆操作的语句必须包含DATA DIRECTORY子句。

集群规模限制：

单个复制组中允许的组成员（MySQL Server实例）最大数量是9个。如果有更多的Server尝试加入该集群时，其连接请求将被拒绝。该限制数量是通过已有的测试案例和基准测试中得出的一个安全边界，在这个安全边界中，集群能够安全、可靠、稳定地运行在一个局域网中。

事务大小限制：

如果单个事务产生的消息内容足够大，以致于无法在5秒的时间内通过网络在组成员之间复制消息，则可能会怀疑该节点失败了，然后将其驱逐出集群，仅仅是因为他们正在忙于处理事务。大型事务还会由于内存分配问题而导致系统变慢。为避免这些问题，请使用以下缓解措施：

如果由于大消息而发生不必要的驱逐，使用系统变量group_replication_member_expel_timeout留出更多时间，以驱逐怀疑失败的节点。在最初的5秒检测期之后，最多可以等待一个小时，然后才能将可疑节点从集群中驱逐出去。从MySQL 8.0.21开始，默认情况下再允许5秒。

在可能的情况下，尝试限制事务的大小，然后再由组复制对其进行处理。例如，将与LOAD DATA一起使用的文件拆分为较小的块。

使用系统变量group_replication_transaction_size_limit可以指定集群接受的最大事务大小。在MySQL 8.0中，该系统变量的默认最大事务大小为150000000字节（约143 MB）。超过此大小的事务将回滚，并且不会发送到组复制的组通信系统（GCS）进行分发。处理事务所花费的时间与其大小成正比，根据需要该集群允许的最大消息大小来调整此变量的值。

使用系统变量group_replication_compression_threshold指定消息大小，在此消息大小之上进行压缩。该系统变量的默认值为1000000字节（1 MB），因此会自动压缩大消息。当组复制的组通信系统（GCS）接收到group_replication_transaction_size_limit设置允许但超过group_replication_compression_threshold设置的消息时，将执行压缩。

使用系统变量group_replication_communication_max_message_size可以指定消息大小，在该消息大小之上可以分段。该系统变量的默认值为10485760字节（10 MiB），因此大型事务消息会自动分段。如果压缩的消息仍超过group_replication_communication_max_message_size限制，则GCS会在压缩后执行分段操作。为了使复制组使用碎片，所有组成员必须在MySQL 8.0.16或更高版本上，并且该组使用的“组复制”通信协议版本必须允许碎片。

通过为相关系统变量指定零值，可以停用最大事务大小，消息压缩和消息碎片。如果停用了所有这些保护措施，则复制组的节点上的应用线程可以处理的消息的大小上限是该节点的slave_max_allowed_packet系统变量的值，该变量的默认最大值为1073741824字节（1 GB）。当接收节点尝试处理此消息时，超出此限制的消息将失败。组成员可以发起并尝试发送到该集群的消息的大小上限为4294967295字节（约4 GB）。这是组通信引擎接受的用于组复制（XCom，Paxos变体）的数据包大小的硬限制，它在GCS处理完消息后接收消息。当发起节点尝试广播超过此限制的消息时，该消息将传递失败。