

## 第十节：组复制常见问题---基础篇

一个集群中MySQL实例的最大数量是多少？

一个集群最多可以包含9个实例。尝试将另一个实例添加到具有9个节点的集群中会导致拒绝加入请求。从已有的测试案例和基准测试中可以确定此限制是安全的边界，在此范围内，集群可以在局域网上稳定的运行。

集群中的实例之间如何通讯和连接？

集群中的实例通过打开TCP连接来建立集群中的各个实例之间点对点的连接。这些连接仅用于集群中节点之间的内部通信和消息传递。此地址由group\_replication\_local\_address变量配置。

group\_replication\_bootstrap\_group选项有什么作用？

引导节点创建集群并充当初始种子节点。加入该集群的第二个节点需要询问引导该集群的节点动态更改配置，以便将其添加到该集群。

节点需要在两种情况下引导集群。最初创建集群时，或者关闭并重新启动整个集群时。

如何为分布式恢复过程设置凭据？

可以使用CHANGE MASTER TO语句将用户凭据永久设置为group\_replication\_recovery通道的凭据。另外，从MySQL 8.0.21开始，可以在每次启动组复制时在START GROUP\_REPLICATION语句中指定它们。

使用CHANGE MASTER TO设置的用户凭据以纯文本格式存储在实例上的复制元数据库中，但是在START GROUP\_REPLICATION上指定的用户凭据仅保存在内存中，并通过STOP GROUP\_REPLICATION语句或实例关闭将其删除。因此，使用START GROUP\_REPLICATION指定用户凭据有助于保护组复制免遭未经授权访问。但是，此方法与group\_replication\_start\_on\_boot系统变量指定的自动启动组复制不兼容。

可以使用组复制来扩展写负载吗？

不会直接拓展写负载，MySQL组复制是一种无共享的完全复制解决方案，其中集群中的所有实例都复制相同数量的数据。因此，如果事务提交操作的结果是集群中的一个节点将N个字节写入存储，则大约N个字节也将被写入其他节点的存储，因为该事务被复制到各个节点上。

但由于其他节点不必执行与原始节点最初执行事务时相同的处理方式，因此他们可以更快地应用更改。事务以仅用于行转换的格式进行复制，而不必再次重新执行事务（基于行的格式）。

此外，考虑到更改是以基于行的格式传播和应用的，这意味着它们以优化且紧凑的格式被接收，并且与原始节点相比，可能会减少所需的IO操作数。

总而言之，可以通过将无冲突事务分布到集群中的不同节点来扩展处理。如果集群使用多主模式，则可以将无冲突的事务分散到不同的主要节点中，这样就能够一定程度上扩展写负载能力（通过多节点写来扩展一小部分IO操作，能够间接扩展一小部分写负载）。

与复制相比，在相同的工作负载下，组复制是否需要更多的网络带宽和CPU？

预期会有一些额外的负载，因为实例之间为了同步数据需要不断相互交互。很难量化需要多少额外的性能。它还取决于集群的大小（与集群中的九台服务器相比，三台服务器对带宽要求的压力较小）。

内存和CPU占用空间也更大，因为服务器同步数据和集群消息传递要完成更复杂的工作。

是否可以在广域网中部署组复制？

是的，但是每个节点之间的网络连接必须可靠并具有很高的性能。低延迟，高带宽网络连接是实现最佳性能的要求。

如果仅仅网络带宽是一个问题，则可以使用“消息压缩”来降低所需的带宽。但是，如果网络丢数据包，导致重传和更高的端到端延迟，则吞吐量和延迟都会受到负面影响。

ps:当任何组成员之间的网络往返时间（RTT）为5秒或更长时，可能会遇到问题，因为内置的故障检测机制可能会被错误地触发。

如果出现临时连接问题，节点是否会重新加入集群？

这取决于连接问题的原因。如果连接问题是暂时性的，并且重新连接的速度足够快，以至于故障检测器无法意识到，则可能无法将实例从集群中删除。如果是“长”连接问题，则故障检测器最终会怀疑是问题，然后将节点从集群中删除。

从MySQL 8.0开始，有两种设置可用于增加节点保留在集群中或重新加入集群的机会：

`group_replication_member_expel_timeout`增加了从产生怀疑（在最初的5秒检测周期后发生）到驱逐节点之间的时间。最多可以设置1小时的等待时间。从MySQL 8.0.21开始，默认情况下等待时间为5秒。

`group_replication_autorejoin_tries`使节点在被驱逐或多数节点超时后尝试重新加入该集群。节点间隔五分钟进行指定次数的自动重新加入尝试。从MySQL 8.0.21开始，此功能默认情况下处于激活状态，并且该节点进行3次自动重新加入尝试。

如果节点从该集群中被驱逐，并且任何自动重新加入尝试均未成功，则需要再次将其重新加入。换句话说，将节点从集群中显式删除后，需要手动重新加入节点（或让脚本自动执行该操作）。

什么时候将节点从集群中剔除？

- 如果某个组成员变为静默状态（不主动发出组通讯消息，也不应答其他节点发送的组通讯消息），则其他组成员会将其从集群配置中删除。实际场景中，当组成员发生崩溃或网络连接断开时可能会发生这种情况。
- 如果某个组成员在经过了给定的时间（这里指的是系统变量 `group_replication_member_expel_timeout` 指定的时间值）之后还处于静默状态，则它将会被驱逐出集群，集群中的其他节点将会创建一个新的集群配置（新的组成员资格）。

当一个节点明显落后于集群时会发生什么？

没有方法来定义何时从集群中自动驱逐节点的策略。需要找出节点落后的原因，然后进行修复或将其从集群中删除。否则，如果节点太慢以至于触发了流控制，那么整个集群也会变慢。可以根据需要配置流量控制机制。

当怀疑集群中有问题时，是否有专门的节点负责触发重新配置？

该集群中没有负责触发重新配置的特殊节点。

任何节点都可以怀疑存在问题。所有节点都需要（自动）同意给定节点已失败。一个节点负责通过触发重新配置将其从集群中驱逐出去。不能控制或设置哪个节点负责驱逐该节点。

可以使用组复制进行分片吗？

- 组复制的目的是提供高可用的副本集，集群中的每个节点都拥有相同的数据，写操作会在集群中的所有节点上进行同步（即，集群中的不同节点之间的数据是一致的，同一个集群中，只有同一份数据但拥有多个副本）。因此，如果要通过数据分片来扩展数据库的写服务能力，需要将数据分散到多个复制组中，然后，构建一个围绕多个复制组的数据分片架构，其中每个复制组只维护和管理整个数据集的给定分片或分区（例如：4个数据分片，每一个数据分片都使用一个单独的复制组来承载）。这种类型的设置，通常称为“分片集群”。这样就实现了对读写服务能力的线性扩展，且不受限制。

如何在SELinux中使用组复制？

如果启用了SELinux（是否启用可以使用`sestatus -v`命令进行检查），则需要开放由系统变量`group_replication_local_address`配置的组复制通信端口的通讯，以便mysqld进程在启动时能够绑定并监听该端口。要查看MySQL当前允许使用哪些端口，可以执行"`semanage port -l | grep mysqld`"命令进行查看。如果发现配置的组通讯端口不在查询结果中，则需要将其添加到SELinux允许的端口列表中。例如：通过命令"`semanage port -a -t mysqld_port_t -p tcp 33061`"可以将33061端口添加到SELinux允许的端口列表中。

如何将组复制与iptables一起使用？

如果启用了iptables，那么需要在iptables允许访问的规则中开放组复制相关的端口，以便组成员之间能够进行正常的组通信。可以在每个组成员的服务器中执行"`iptables -L`"命令查看当前生效的防火墙规则。如果发现配置的组通讯端口不在查询结果中，则需要将其添加到iptables允许的访问规则中。例如：通过命令"`iptables -A INPUT -p tcp --dport 33061 -j ACCEPT`" 可以将33061端口添加到被允许的防火墙规则中。

如何恢复组成员使用的复制通道中继日志？

组复制使用的复制通道的行为与主从复制使用的复制通道相同（组复制是基于主从复制的基础架构上实现的），因此组复制依赖于中继日志。如果修改了系统变量`relay_log`的值、或者未显式设置该系统变量的值时主机名发生了变化，那么组复制通道大概率可能出现错误。如果出现这种情况，可以使用如下两种方式尝试解决：

- \* 停止出现问题的组成员的组复制通道，然后，将旧的中继日志索引文件的内容预先添加到新的中继日志索引文件中，然后，重新启动组复制通道（这种方法其实是在主从复制拓扑中，中继日志出现名称问题时的解决方法，在组复制中，建议使用第二种方法）。

- \* 使用`STOP GROUP_REPLICATION`语句停止组复制通道，然后直接执行`START GROUP_REPLICATION`语句来重新启动组复制通道。组复制插件会自动重新创建新的`group_replication_applier`通道。

为什么组复制使用两个绑定地址？

组复制使用两个绑定地址，以便在客户端用于与节点通信的SQL地址和组成员在内部用于通信的`group_replication_local_address`之间分配网络流量。例如，假定实例具有分配给网络地址203.0.113.1和198.51.100.179的两个网络接口。在这种情况下，可以通过设置`group_replication_local_address = 203.0.113.1: 33061`将203.0.113.1:33061用作内部组网络地址。然后可以将198.51.100.179用作主机名，并将3306用作端口。然后，客户端SQL应用程序将在198.51.100.179:3306连接到该节点。这可以在不同的网络上配置不同的规则。同样，可以将内部组通信与用于客户端应用程序的网络连接分开，提高安全性。

组复制如何使用网络地址和主机名？

组复制使用节点之间的网络连接，因此，其功能直接受到配置主机名和端口的方式的影响。例如，组复制的分布式恢复过程使用server的主机名和端口创建到现有组成员的连接。当节点加入集群时，它将使用`performance_schema.replication_group_members`中列出的网络地址信息接收组成员信息。该表中列出的节点之一被选为从集群到加入节点的丢失数据的提供者。

使用主机名配置的任何值（例如SQL网络地址或集群种子地址）都必须是标准名称，并且集群的每个节点都可以解析。可以通过DNS或正确配置的`/etc / hosts`文件或其他本地进程来确保这一点。如果要在实例上配置`MEMBER_HOST`值，请在将其加入集群之前使用实例上的`--report-host`选项指定它。

ps:该变量分配的值将直接使用，并且不受`skip_name_resolve`系统变量的影响。

要在实例上配置`MEMBER_PORT`，请使用`report_port`系统变量指定它。

为什么实例上的自增量设置会更改？

在节点上启动组复制时，`auto_increment_increment`的值将更改为`group_replication_auto_increment_increment`的值（默认值为7），而`auto_increment_offset`的值将更改为server ID。停止组复制时，将更改还原。这些设置避免为组成员上的写入选择重复的自增值，这会导致事务回滚。组复制的默认自增值为7，表示可用值与复制组的允许最大大小（9个节点）之间的平衡。

仅当`auto_increment_increment`和`auto_increment_offset`各自的默认值均为1时，才进行还原更改。如果已将其值从默认值修改，则组复制不会更改它们。从MySQL 8.0开始，当组复制处于只有一个节点写入的单主模式下时，系统变量也不会被修改。

如何找到primary节点？

如果该集群以单主模式运行，则找出哪个节点是primary节点会很有用。