

第十六节：组复制升级---进阶篇

1.概述：

升级集群节点的基本过程与升级独立实例的过程相同，在就地升级或逻辑升级之间进行选择取决于集群中存储的数据量。通常，就地升级速度更快，因此建议使用。

在线升级集群的过程中，为了最大程度地提高集群可用性，可能需要同时运行具有不同MySQL Server版本的节点。组复制的兼容性策略能够在升级过程中将运行不同版本MySQL的节点安全地合并到同一集群中。这些策略可能会影响升级集群内节点的顺序，具体取决于集群如何配置。

2.在一个集群中兼容不同版本的节点：

2.1概述：

根据组复制插件绑定的MySQL Server版本对组复制进行版本控制。例如，如果节点运行的是MySQL 5.7.26，则该版本是“组复制”插件的版本。

使用如下语句在集群节点上检查MySQL Server的版本：

```
SELECT MEMBER_HOST, MEMBER_PORT, MEMBER_VERSION FROM
performance_schema.replication_group_members;
+-----+-----+-----+
| member_host | member_port | member_version |
+-----+-----+-----+
| node1      | 3306        | 8.0.17         |
+-----+-----+-----+
```

为了获得最佳的兼容性和性能，集群的所有节点应运行相同版本的MySQL Server，应运行相同的组复制版本。但是，在线升级集群的过程中，为了最大程度地提高集群的可用性，可能需要同时运行具有不同MySQL Server版本的节点。根据不同MySQL版本之间所做的调整，在这种情况下可能会遇到兼容性引发的问题。

例如，如果某个功能在新版本已被弃用，则一个集群内可能会出现导致依赖于已弃用的功能的高版本节点写入数据失败。相反，当运行旧版MySQL的集群中时，写入运行低版本MySQL的节点可能会导致缺少高版本引入的功能的低版本节点出现写入失败的问题。

为防止出现这些问题，组复制兼容性策略可以安全地在同一集群中组合运行不同版本MySQL的节点。节点应用这些策略来决定是正常加入集群还是以只读模式加入，还是不加入集群，具体取决于哪种选择会导致加入节点和集群现有节点的安全运行。在滚动升级时，每个节点必须离开集群进行升级，然后以其新版本重新加入该集群。此时，集群中的节点需要为新版本做对应的策略匹配（该策略可能与组成员升级之前应用的策略不同）。

作为DBA，可以通过适当配置并执行START GROUP_REPLICATION语句，让节点尝试加入集群。在尝试将其添加到集群后，将由加入节点自己决定加入或不加入该集群或以只读模式加入该集群。加入节点将接收当前组成员的MySQL Server版本的信息，评估其自身与那些节点的兼容性，并应用自己的MySQL Server版本中使用的策略（而不是现有节点使用的策略）来确定是否为兼容。

尝试加入集群时加入节点所应用的兼容性策略如下：

如果加入节点所运行的MySQL Server版本低于现有组成员所运行的最低版本，则该节点不会加入该集群。

如果加入节点运行与现有组成员所运行的最低版本相同的MySQL Server版本，则该节点通常会加入该集群。

节点加入集群后，如果运行的MySQL Server版本高于现有组成员运行的最低版本，则该节点仍处于只读模式。仅当集群在多主模式下运行时，此行为才有所不同，因为在单主模式下运行的集群中，新添加的节点默认为只读。

运行MySQL 8.0.17或更高版本的节点在检查其兼容性时会考虑该发行版的补丁版本。运行MySQL 8.0.16或更低版本或MySQL 5.7的节点仅考虑主要版本。

例如，如果有一个集群均运行在MySQL版本为8.0.13的节点上：

运行MySQL 5.7的节点不允许加入集群。

运行MySQL 8.0.16的节点可以正常加入（因为集群只考虑该节点的主要版本）。

运行MySQL 8.0.17的节点会加入，但仍处于只读模式（因为既要考虑主要版本也要考虑次要版本）。

在MySQL 5.7.27之前版本的加入节点会检查所有组成员，以发现自己的MySQL Server主版本是否较低。对于任何集群正在运行MySQL 8.0的节点，他们将无法通过此检查，即使该集群已经具有运行MySQL 5.7的其他节点，也无法加入该集群。从MySQL 5.7.27开始，加入节点仅检查运行最低主要版本的组成员，因此他们可以加入存在其他MySQL 5.7机器以上版本节点的混合集群。

在节点使用不同MySQL server版本的多主模式集群中，组复制自动管理运行MySQL 8.0.17或更高版本的节点的读写状态。如果节点离开该集群，则运行当前最低版本的节点将自动设置为读写模式。当使用group_replication_switch_to_multi_primary_mode () UDF将以单主模式运行的集群更改为以多主模式运行时，组复制会自动将节点设置为正确的模式。如果节点运行的MySQL server版本高于集群中存在的最低版本的节点，则节点将自动置于只读模式，而运行最低版本的节点将处于读写模式。

2.2升级集群节点：

在线升级过程中，如果集群处于单主模式，则当前未退出集群进行升级的所有实例均会像以前那样运行。如果要求primary节点在整个过程中保持不变（除非它本身正在升级），则必须首先将所有secondary节点升级到高于或等于primary节点的版本，然后再升级primary节点。除非该节点正在运行集群中最低的MySQL Server版本，否则它不能保留为primary节点。升级primary节点后，可以使用group_replication_set_as_primary () UDF将其重新指定为primary节点。

如果该集群处于多主模式，则在升级过程中可用于执行写入操作的online节点会更少，因为升级后的节点在升级后会以只读模式加入集群。从MySQL 8.0.17开始，这适用于补丁程序版本之间的升级，而对于较低的发行版，仅适用于主要版本之间的升级。当所有节点都已从MySQL 8.0.17升级到同一版本时，它们都会自动变回读写模式。对于早期的低版本，必须在升级后将每个primary节点的super_read_only手动设置为OFF。

ps:对于新旧版本的比较而言，从MySQL 8.0.17开始，比较时需要考虑次要版本号，而对于8.0.16及其之前的版本，做版本比较时只考虑主要版本号。

如果必须在紧急情况下回滚升级或为集群添加额外的节点，则可以允许节点加入在线集群，尽管该节点运行的MySQL Server版本低于集群内其他节点使用的最低版本。在这种情况下，可以使用组复制系统变量group_replication_allow_local_lower_version_join覆盖常规兼容性策略。要注意，将该选项设置为ON不会使新节点与该集群兼容，并且允许它加入该集群而没有任何措施防止现有节点的不兼容行为。因此，只能在特定情况下谨慎使用该变另，并且必须采取其他预防措施，以避免新节点由于正常的集群活跃事务而失败。

2.3组复制通信协议版本：

MGR使用的组复制通信协议版本可能与集群内节点的MySQL Server版本不同。要检查集群的通信协议版本，在集群内任意节点上执行以下语句：

```
SELECT group_replication_get_communication_protocol();
```

返回值显示可以加入该集群并使用该集群的通信协议的最旧的MySQL Server版本。MySQL 5.7.14的版本允许压缩消息，而MySQL 8.0.16的版本也支持消息分段。

`group_replication_get_communication_protocol ()` UDF返回该集群支持的最低MySQL版本，该最低版本可能与传递给`group_replication_set_communication_protocol ()` UDF的版本号以及安装在使用该节点集群上的MySQL Server版本不同，例如：传递给该函数的值为8.0.17，最终生效的是8.0.16。。

当将集群的所有节点升级到新的MySQL Server版本时，如果仍然需要允许早期低版本的节点加入，则组复制通信协议版本不会自动升级。如果不需要支持较旧的节点并希望允许升级的节点使用高版本的通信协议，则在升级后，使用`group_replication_set_communication_protocol ()` UDF升级通信协议，并指定新版本的MySQL Server版本。

3.组复制离线升级：

要执行组复制集群的离线升级，需要从集群中删除每个节点，执行该节点的升级，然后照常重新启动该集群。在多主集群中，可以按任意顺序关闭节点。在单主集群中，先关闭每个secondary节点，然后再关闭primary节点。

集群离线后，按照正常升级MySQL实例升级所有节点。所有节点升级后，重新启动集群。

如果在复制组的所有节点离线时对其进行升级，然后重新启动该集群，则这些节点将使用新发行版的“组复制”通信协议版本加入集群，从而成为该集群的通信协议版本。如果需要允许早期低版本中的节点加入，则可以使用`group_replication_set_communication_protocol ()` UDF降级通信协议版本，并指定具有最旧安装版本的预期组成员的MySQL Server版本。

4.在线升级组复制：

在线升级组复制集群时，应注意以下几点：

- 1.无论升级集群的方式如何，一定要禁止对组成员的任何写操作，直到集群升级完毕为止。
- 2.当节点停止时，`super_read_only`变量会自动设置为on，但此变更不会持久化到配置文件中。
- 3.当MySQL 5.7.22或MySQL 8.0.11尝试加入运行MySQL 5.7.21或更低版本的集群时，由于MySQL 5.7.21不会发送其`lower_case_table_names`的值，所以它无法加入该集群。

4.1升级组复制节点：

升级集群中所有节点的方法都是通用的，但要注意，集群运行在单主模式还是多主模式，会影响组成员升级并重新加入集群的顺序。

升级组成员的过程包括：将其从集群中剔除，按照选择的升级成员的方法执行升级，然后重新将升级后的节点加入到集群中。对单主模式集群中的节点执行升级时，建议先依次升级只读节点，最后再升级读写节点，如果读写节点需要先于只读节点升级，则需要先选择一个旧版本的其他节点作为新的primary节点。所以，为减少工作量，通常建议读写节点最后升级。

升级步骤：

使用客户端登录到将要执行升级的节点上并执行`STOP GROUP_REPLICATION`。在继续操作之前，需通过`Replication_group_members`表来确保集群内节点的状态为OFFLINE。

禁止组复制自动启动，以便可以在升级和配置后安全地以手动的方式将该节点加入集群，而无需通过设置`group_replication_start_on_boot = 0`在升级后该节点自动加入该集群。

ps:如果升级的节点具有`group_replication_start_on_boot = 1`，则该节点可以在升级完成后自动加入集群，但可能导致问题。例如，如果升级失败并且实例再次重新启动，则升级失败的实例可以加入该集群。

停止该节点，例如使用`mysqladmin shutdown`或`SHUTDOWN`语句。该集群中的任何其他节点将继续运行。

使用原地（使用mysql_upgrade命令升级数据字典方式）或使用预先安装配置好新版本Server进行逻辑导数的方法升级。升级后执行重新启动Server时，由于系统变量group_replication_start_on_boot设置为0，不会自动启动组复制并重新加入集群，所以需要人工操作重新加入集群。

在节点上执行MySQL升级过程后，必须将group_replication_start_on_boot设置为1，以确保组复制在重启后正确启动。

连接到升级后的节点并执行START GROUP_REPLICATION。这会将该节点重新加入该集群。组复制元数据在升级的实例上已经就位，因此通常不需要重新配置组复制。

新加入集群升级后的节点必须要追赶集群中最新的事务，当它追赶上了集群中的最新事务之后，它就会成为这个集群的online节点。

ps：升级Server所需的时间越长（即，离开集群的时间就越长），该Server与集群中的数据差异可能就越大，在将其重新添加回集群时，所需的时间也就越多（因为可能有更多的事务需要追赶）。

当升级的节点加入具有运行早期MySQL Server版本的任何节点的集群时，升级的节点将以super_read_only = on加入集群。这样可以确保在所有节点都运行较新版本之前，不对升级后的节点进行写操作。在多主模式集群中，当升级成功完成并且该集群准备好处理业务数据时，必须将打算用作可写的primary节点设置为读写模式。从MySQL 8.0.17开始，当一个集群的所有节点都升级到相同的发行版时，它们都会自动变回读写模式。对于早期版本，必须手动将每个节点设置为读写模式。

连接到每个节点并执行：

```
SET GLOBAL super_read_only=OFF;
```

4.2在线升级组复制方法：

4.2.1集群内滚动升级：

集群内滚动升级：指的是组成员逐个剔除出集群、执行原地升级（不需要迁移数据，不需要更换服务器），升级完成之后，再重新加入集群。在执行升级的过程中，集群全程能够对外提供读写服务，但被剔除出集群并执行升级的节点在执行升级的过程中不承载任何工作负载（不提供只读或读写服务），当节点升级完成之后，重新加入集群时，如果集群内存在着更低版本号节点，则它会以只读方式重新加入集群（此时只是提供只读服务）。对于新加入的节点后续是否需要从只读模式变更为读写模式，则需要看集群运行在单主模式还是多主模式。

此方法非常适合单主集群。当集群以单主模式运行时，如果要求primary节点在整个过程中保持不变（除非它本身正在升级），则它应该是要升级的最后一个节点。除非该primary节点正在运行集群中最低的MySQL Server版本，否则它不能保留为primary节点。升级primary节点后，可以使用group_replication_set_as_primary () UDF将其重新指定为primary节点。如果不介意哪个节点是primary节点，则可以按任何顺序升级集群内的节点。该集群会在必要时从运行最低MySQL Server版本的节点中选择一个新的primary节点。

对于以多主模式运行的集群，在滚动升级期间，primary节点数量会减少，从而导致写入可用性降低。这是因为如果节点加入的集群在运行的MySQL Server版本高于现有集群所运行的最低版本时，它将自动保持为只读模式（super_read_only = ON）。运行MySQL 8.0.17或更高版本的节点在进行检查时会考虑该发行版的补丁版本，但是运行MySQL 8.0.16或更低版本或MySQL 5.7的节点仅会考虑主版本。当所有节点都已从MySQL 8.0.17升级到同一版本时，它们都会自动变回读写模式。对于早期的低版本，必须在升级后为每个primary节点手动设置super_read_only = OFF。

4.2.2滚动迁移升级

迁移滚动升级：指的是组成员逐个剔除出集群、执行升级之后，不重新加入原来的集群中，而是使用升级后的节点创建新集群。对于以多主模式运行的集群，在此过程中读写节点的数量会逐渐减少，导致写可用性降低。但对于以单主模式运行的集群，这不会影响集群的写可用性（但在集群最后升级读写节点时，应用需要在新旧集群之间执行一次写请求切换，此时应用会受到影响）。

在对集群执行升级的过程中，因为运行旧版本的集群一直处于在线状态（持续对外提供读写和只读服务），所以更新完版本的节点组成的新的集群必须要追赶升级过程中，旧集群中新写入的任何事务。因此，需要在新集群中选定一个节点作为从库，与旧集群中的读写节点之间建立一个异步的主从复制通道来追赶最新的数据。为避免通过异步复制通道追赶数据出现意外，需要保证新旧集群之间关于组复制和主从复制相关的系统配置参数完全一致。对于以单主模式运行的集群，新集群中作为从库的节点必须是新集群中的读写节点，对于多主模式运行的集群，新集群中作为从库的节点可以是新集群中的任意节点（读写节点）。

过程如下：

从运行较旧版本的原始集群中逐个删除节点。

升级在节点上运行的实例版本，可以采用就地升级或备份升级的方式。

使用升级的节点创建新集群。在这种情况下，需要在每个节点上配置一个新的集群名（因为旧的集群仍在运行并使用旧的名称），引导一个初始的升级节点作为primary节点，然后添加其余的升级完成的节点到集群中。

在旧集群和新集群之间设置异步复制通道，将较旧的primary节点配置为充当异步复制源服务器，并将新的集群primary节点配置为基于GTID复制的副本。

在将应用程序重定向到新集群之前，必须确保新集群具有适当数量的组成员，以便新集群在有节点发生故障时可以正常应对。可以通过在新集群的任意节点中通过 `performance_schema.replication_group_members` 表查询组成员视图来查看旧集群和新集群的大小，在确保该信息之后，可以在旧集群中阻塞数据写入（需要观察新旧集群之间的异步复制延迟，延迟不大时可以执行此步骤），并等待新集群中追赶旧集群中的最新数据，直到新集群追赶上旧集群的所有数据为止，然后，切换应用程序到新集群中，并删除新旧集群之间的异步复制连接，最后，升级所有旧版本的组成员，升级完成之后，将其逐个加入到新集群中。

PS：迁移滚动升级的方案，组成员在升级完成之后，节点会组成新集群，服务器（主机）还是原来的。

4.2.3副本滚动升级：

副本滚动升级：与"集群内滚动升级"和"迁移滚动升级"两种方案比起来，组成员不需要做剔除出集群的操作，而是使用备份工具，将集群中的数据副本完整拷贝到一组新的服务器中，逐个执行升级，并创建新集群，在执行升级期间，由于旧集群持续在线对外提供服务，新旧集群之间的增量数据，需要通过在新旧集群之间建立异步的复制通道进行数据同步。当新集群追赶上旧集群中的最新数据时，将应用程序的访问切换到新集群。对于以多主模式运行的集群，升级过程中读写节点的数量不会减少，所以，写的可用性不会受到影响（因此，这种升级方式非常适合运行多主模式的集群）。对于以单主模式运行的集群，写可用性同样不会受到影响。

由于此方法依赖于异步复制通道，该通道用于将数据从一个集群复制到另一集群，因此需要保证就集群的master节点和新集群的slave节点在MGR和一部复制相关的参数模板保持一致。对于以单主模式运行的集群，与旧集群的异步复制连接必须将数据发送到新集群中的primary节点，对于多主集群，异步复制通道可以连接至任何primary节点。

步骤如下：

使用新版本Server程序包在新的服务器中进行初始化安装，需要确保新版本的节点数量在有节点发生故障时能够正常应对。注意，此时不需要在新版本的服务器上创建集群，只需要使用新版本程序包初始化安装数据库Server，并确保Server能够正常启动即可。

从原集群中的某个节点中备份现有数据（完整备份）

使用备份数据恢复到准备创建新集群的服务器中，执行原地升级。

必须将备份还原到与备份相同的MySQL版本，然后执行就地升级。

用升级到新版本的Server创建一个新集群，要注意，由于旧集群正在运行，因此需要给新集群命名一个新的集群名称，并使用第一个升级完成的Server来引导新的集群，后续升级完成的Server加入新集群即可。

在旧集群和新集群之间设置异步复制通道，将较旧的primary节点配置为充当异步复制源实例，并将新的组成员配置为基于GTID的副本。

一旦新集群中的数据相较于旧集群的复制延迟足够小时，就可以执行应用的切换，然后，删除新旧组之间的异步复制通道。

PS：副本滚动升级方案，组成员在升级完成之后，组成员和服务（主机）都是全新的。

5.使用xtrabackup升级组复制

作为升级方法的一部分，可以使用xtrabackup将数据从组成员复制并还原到新集群节点。但是，不能使用此技术直接将备份从运行旧版MySQL的节点还原到运行新版MySQL的节点。解决方案是将备份还原到一个新实例，该实例运行与从中获取备份的成员相同版本的MySQL，然后升级该实例。该过程包括：

使用xtrabackup从旧集群的节点中进行备份。

部署一个新的实例，该实例必须与执行备份的旧节点运行相同版本的MySQL。

使用xtrabackup将备份从旧实例还原到新实例。

在新实例上升级MySQL。

重复此过程创建适当数量的新实例，以便能够处理故障转移。然后根据“组复制在线升级方法”将实例加入新集群。