

16.主从复制

通过复制，可以将来自一个MySQL数据库服务器（主服务器）的数据复制到一个或多个MySQL数据库服务器（从服务器）。复制在默认情况下是异步的，因此从站不需要永久连接以接收来自主站的更新。这意味着可以通过长途连接进行更新，甚至可以通过临时或间歇连接（如拨号服务）进行更新。根据配置，可以复制数据库中的所有数据库，选定数据库甚至选定的表。

MySQL中复制的优点包括：

- 横向扩展解决方案 - 将负载分散到多个slaves以提高性能。在此环境中，所有写入和更新都必须在master上进行。然而，阅读可能发生在个或多个slaves身上。这种模式可以提高写入的性能（因为主机专门用于更新），同时显着提高越来越多的从机的读取速度。
- 数据安全性 - 因为数据被复制到从机，从机可以暂停复制过程，所以可以在从机上运行备份服务，而不会破坏相应的主机数据。
- 分析 - 可以在主站上创建实时数据，而信息的分析可以在从站上进行，而不会影响主站的性能。
- 远程数据分发 - 您可以使用复制为远程站点创建本地数据副本，而无需永久访问主数据。

有关如何在这种情况下使用复制的信息，请参见第16.3节“复制解决方案”。

MySQL 5.7支持不同的复制方法。传统的方法是基于从主站的二进制日志中复制事件，并要求主站和从站之间的日志文件和位置同步。基于全局事务标识符（GTID）的更新方法是事务性的，因此不需要使用这些文件中的日志文件或位置，这大大简化了许多常见的复制任务。只要在主站上提交的所有事务也被应用在从站上，使用GTID的复制保证主站和从站之间的一致性。有关MySQL中GTID和基于GTID的复制的更多信息，请参见第16.1.3节“使用全局事务标识符进行复制”。有关使用二进制日志文件基于位置的复制的信息，请参见第16.1节“配置复制”。

MySQL中的复制支持不同类型的同步。原始类型的同步是单向异步复制，其中一台服务器充当主服务器，另外一台或多台服务器充当从服务器。这与作为NDB集群特征的同步复制相反（参见第21章，MySQL NDB集群7.5和NDB集群7.6）。在MySQL 5.7中，除了内置的异步复制，还支持半同步复制。使用半同步复制，在返回执行事务的会话之前在主模块上执行的提交，直到至少一个从服务器确认它已经接收并记录事务的事件;请参见第16.3.9节“半同步复制”。MySQL 5.7还支持延迟复制，使得从服务器有意地在主服务器上至少延迟一段指定的时间;请参见第16.3.10节“延迟复制”。对于需要同步复制的情况，请使用NDB集群（请参阅第21章，MySQL NDB集群7.5和NDB集群7.6）。

有许多解决方案可用于设置服务器之间的复制，最佳的使用方法取决于数据的存在以及您正在使用的引擎类型。有关可用选项的更多信息，请参见第16.1.2节“设置基于二进制日志文件的位置复制”。

有两种复制格式的核心类型：复制整个SQL语句的基于语句的复制（SBR）和仅复制已更改行的基于行的复制（RBR）。您也可以使用第三种混合基复制（MBR）。有关不同复制格式的更多信息，请参见第16.2.1节“复制格式”。

复制通过许多不同的选项和变量进行控制。有关更多信息，请参见第16.1.6节“复制和二进制日志记录选项和变量”。

您可以使用复制来解决许多不同的问题，包括性能，支持不同数据库的备份，以及作为缓解系统故障的更大解决方案的一部分。有关如何解决这些问题的信息，请参见第16.3节“复制解决方案”。

有关在复制期间如何处理不同数据类型和语句的说明和提示，包括复制功能，版本兼容性，升级以及潜在问题及其解决方法的详细信息，请参见第16.4节“复制注释和提示”。有关MySQL复制新手经常问到的一些问题的答案，请参见第A.13节“MySQL 5.7 FAQ：复制”。

有关复制实现的详细信息，复制如何工作，二进制日志的进程和内容，后台线程以及用于决定语句如何记录和复制的规则，请参见第16.2节“复制实现”。

16.1 复制配置

本节介绍如何配置MySQL中可用的不同类型的复制，并包括复制环境所需的设置和配置，包括创建新复制环境的分步说明。本节的主要内容是：

- 有关使用二进制日志文件位置设置两个或更多服务器进行复制的指南，请参见第16.1.2节“设置基于二进制日志文件的位置复制”，处理服务器的配置，并提供在主服务器 和从服务器。
- 有关使用GTID事务设置两个或更多服务器进行复制的指导，请参见第16.1.3节“使用全局事务标识符进行复制”处理服务器的配置。
- 二进制日志中的事件使用多种格式进行记录。这些被称为基于语句的复制（SBR）或基于行的复制（RBR）。第三种类型混合格式复制（MIXED）在适当时自动使用SBR或RBR复制来利用SBR和RBR格式的优点。第16.2.1节“复制格式”中讨论了不同的格式。
- 有关适用于复制的不同配置选项和变量的详细信息，请参见第16.1.6节“复制和二进制日志记录选项和变量”。
- 一旦开始，复制过程应该只需要很少的管理或监控。但是，有关可能需要执行的常见任务的建议，请参见第16.1.7节“常见复制管理任务”。

16.1.1 基于二进制日志文件位置的复制配置概述

本节介绍基于二进制日志文件定位方法的MySQL服务器之间的复制，其中作为主服务器的MySQL实例（数据库的来源发生更改）将更新和更改作为“事件”写入二进制日志。根据记录的数据库更改，二进制日志中的信息以不同的日志记录格式存储。从站被配置为从主站读取二进制日志，并执行从站本地数据库的二进制日志中的事件。

每个从机接收到二进制日志的全部内容的副本。从机决定执行二进制日志中的哪些语句是责任。除非另外指定，否则主站二进制日志中的所有事件都将在从站上执行。如果需要，可以将从站配置为仅处理适用于特定数据库或表的事件。

重要 您无法将主设备配置为仅记录特定事件。

每个从机都保存二进制日志坐标的记录：从主机读取并处理的文件名称和位置。这意味着多个从站可以连接到主站并执行同一个二进制日志的不同部分。由于从站控制这个过程，因此可以将各个从站与服务器连接和断开连接，而不会影响主站的操作。另外，由于每个从站都记录二进制日志中的当前位置，因此可能会断开从站的连接，重新连接，然后继续处理。

主站和每个从站必须配置一个唯一的ID（使用server-id选项）。另外，每个从站都必须配置有关主站名称，日志文件名以及该文件中的位置的信息。这些细节可以在MySQL会话中使用从站上的CHANGE MASTER TO语句进行控制。详细信息存储在从属主信息库中，可以是文件或表（请参见第16.2.4节“复制中继和状态日志”）。

16.1.2 设置基于二进制日志文件位置的复制

本节介绍如何设置MySQL服务器以使用基于二进制日志文件位置的复制。有许多不同的方法来设置复制，使用的方法确切方法取决于您如何设置复制，以及您的主数据库中是否已有数据。

有一些通用的任务是所有设置都通用的：

- 在主服务器上，您必须启用二进制日志记录并配置唯一的服务器ID。这可能需要重新启动服务器。请参见第16.1.2.1节“设置复制主配置”。
- 在要连接到主站的每个从站上，必须配置唯一的服务器ID。这可能需要重新启动服务器。请参见第16.1.2.5.1节“设置复制从站配置”。

- 或者，在读取二进制日志进行复制时，为主服务器创建一个单独的用户，以便在与主服务器进行身份验证时使用。请参见第16.1.2.2节“创建复制用户”。
- 在创建数据快照或启动复制过程之前，在主服务器上您应该记录二进制日志中的当前位置。配置从站时需要这些信息，以便从站知道二进制日志中的哪个位置开始执行事件。请参见第16.1.2.3节“获取复制主二进制日志坐标”。
- 如果您已经拥有主设备上的数据并想使用它来同步从设备，则需要创建数据快照以将数据复制到从设备。您正在使用的存储引擎对创建快照的方式有影响。当您使用MyISAM时，必须停止处理master上的语句以获得读锁，然后获取其当前的二进制日志坐标并转储其数据，然后才允许master继续执行语句。如果不停止语句的执行，数据转储和主状态信息将不匹配，从而导致从属数据库不一致或损坏。有关复制MyISAM主节点的更多信息，请参见第16.1.2.3节“获取复制主二进制日志坐标”。如果你使用的是InnoDB，你不需要一个读锁，一个足够长的传输数据的事务就足够了。有关更多信息，请参见第14.19节“InnoDB和MySQL复制”。
- 使用连接到主站的设置来配置从站，例如主机名，登录凭证和二进制日志文件的名称和位置。请参见第16.1.2.5.2节“在从站上设置主站配置”。

注意 设置过程中的某些步骤需要SUPER权限。如果您没有此权限，则可能无法启用复制。

配置基本选项后，选择您的方案：

- 要设置全新安装主站和不包含数据的从站的复制，请参见第16.1.2.5.3节“设置新主站和从站之间的复制”。
- 要使用现有MySQL服务器中的数据设置新主数据库的复制，请参见第16.1.2.5.4节“使用现有数据设置复制”。
- 要将复制从节点添加到现有的复制环境，请参见第16.1.2.6节“将复制节点添加到复制环境”。

在管理MySQL复制服务器之前，请阅读完整章节并尝试第13.4.1节“用于控制主服务器的SQL语句”和第13.4.2节“用于控制从服务器的SQL语句”中提到的所有语句。还要熟悉第16.1.6节“复制和二进制日志记录选项和变量”中介绍的复制启动选项。

16.1.2.1 设置复制主配置

要将主服务器配置为使用基于二进制日志文件位置的复制，必须启用二进制日志记录并建立唯一的服务器ID。如果尚未完成，则需要重新启动服务器。

二进制日志记录必须在主服务器上启用，因为二进制日志是从主服务器复制到其从服务器的基础。如果使用log-bin选项在主服务器上未启用二进制日志记录，则复制是不可能的。

复制组中的每个服务器都必须配置唯一的服务器标识。此ID用于标识组中的单个服务器，并且必须是1到（ 2^{32} ）-1之间的正整数。如何组织和选择数字是您的选择。

要配置二进制日志和服务器ID选项，请关闭MySQL服务器并编辑my.cnf或my.ini文件。在配置文件的[mysqld]部分中，添加log-bin和server-id选项。如果这些选项已经存在，但已被注释掉，请取消注释选项并根据您的需要进行更改。例如，要使用mysql-bin的日志文件名称前缀启用二进制日志记录，并将服务器ID配置为1，请使用以下行：

```
[mysqld]
log-bin=mysql-bin
server-id=1
```

进行更改后，重新启动服务器。

注意

以下选项对此过程有影响：

- 如果您省略server-id（或将其明确设置为默认值0），则主设备将拒绝来自从设备的任何连接。
- 为了在使用InnoDB和事务的复制设置中保持最大的持久性和一致性，您应该在master my.cnf文件中使用innodb_flush_log_at_trx_commit = 1和sync_binlog = 1。
- 确保您的复制主服务器上未启用跳过网络选项。如果网络已禁用，从站无法与主站通信，复制失败。

16.1.2.2 创建复制用户

每个从设备都使用MySQL用户名和密码连接到主设备，所以主设备上必须有一个用户帐户，从设备才能使用该帐户进行连接。任何帐户都可以用于此操作，只要它已被授予REPLICATION SLAVE权限。您可以选择为每个从站创建一个不同的帐户，或者使用每个从站的相同帐户连接到主站。

虽然您不必专门为复制创建帐户，但您应该知道复制用户名和密码以纯文本形式存储在主信息存储库文件或表中（请参见第16.2.4.2节“从属状态日志”）。因此，您可能希望创建一个仅具有复制过程权限的单独帐户，以尽可能减少对其他帐户的危害。

要创建一个新帐户，请使用CREATE USER。要授予此帐户复制所需的权限，请使用GRANT语句。如果您仅为复制目的创建帐户，则该帐户只需要REPLICATION SLAVE权限。例如，要设置一个新用户repl，它可以连接到mydomain.com域中的任何主机进行复制，请在主服务器上发出以下语句：

```
mysql> CREATE USER 'repl'@'%mydomain.com' IDENTIFIED BY 'slavepass';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%mydomain.com';
```

有关操作用户帐户的语句的更多信息，请参见第13.7.1节“帐户管理语句”。

16.1.2.3 获取复制主二进制日志坐标

要将从站配置为在正确的位置启动复制过程，需要主站的二进制日志中的当前坐标。

如果主服务器以前没有启用二进制日志记录，SHOW MASTER STATUS或mysqldump --master-data显示的日志文件名和位置值是空的。在这种情况下，稍后在指定从站的日志文件和位置时需要使用的值是空字符串（"）和4。

如果以前master已经是二进制日志记录，那么使用这个过程来获取master二进制日志坐标：

警告

此过程使用FLUSH TABLES WITH READ LOCK，它阻止InnoDB表的COMMIT操作。

1. 通过与命令行客户端连接，在主服务器上启动一个会话，并通过执行FLUSH TABLES WITH READ LOCK语句来清除所有表并阻止写入语句：

```
mysql> FLUSH TABLES WITH READ LOCK;
```

警告

保持您发出FLUSH TABLES语句的客户端运行，以便读取锁定保持有效。如果您退出客户端，锁定被释放。

2. 在master上的不同会话中，使用SHOW MASTER STATUS语句来确定当前二进制日志文件的名称和位置：

```
mysql > SHOW MASTER STATUS;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
mysql-bin.000003	73	test	manual,mysql

“File”列显示日志文件的名称，“Position”列显示文件中的位置。在这个例子中，二进制日志文件是mysql-bin.000003，位置是73。记录这些值。你以后需要它们，当你设置从服务器。它们表示从站应该开始处理来自主站的新更新的复制坐标。

您现在拥有所需的信息，可以使从站从正确位置的二进制日志开始读取，以开始复制。

下一步取决于您是否有主数据。选择以下选项之一：

- 如果您在启动复制之前需要与从属服务器同步现有的数据，请保持客户端正常运行，以便保持锁定位置。这防止了进一步的更改，以便复制到从站的数据与主站同步。继续第16.1.2.4节“选择数据快照的方法”。
- 如果您正在设置新的主从复制组，则可以退出第一个会话以释放读锁定。有关如何继续操作，请参见第16.1.2.5.3节“设置新主站和从站之间的复制”。

16.1.2.4 为数据快照选择一种方法

如果主数据库包含现有数据，则需要将这些数据复制到每个从站。有不同的方法来从主数据库转储数据。以下各节介绍可能的选项。

要选择转储数据库的适当方法，请在以下选项中进行选择：

- 使用mysqldump工具创建所有要复制的数据库的转储。这是推荐的方法，特别是在使用InnoDB时。
- 如果您的数据库存储在二进制可移植文件中，则可以将原始数据文件复制到从站。这可以比使用mysqldump并在每个slave上导入文件更高效，因为它会在重播INSERT语句时跳过更新索引的开销。对于像InnoDB这样的存储引擎，这是不推荐的。

16.1.2.4.1 使用mysqldump创建数据快照

要在现有主数据库中创建数据的快照，请使用mysqldump工具。数据转储完成后，在开始复制过程之前将此数据导入到从站。

以下示例将所有数据库转储到名为dbdump.db的文件，并包含--master-data选项，该选项会自动追加从属服务器上所需的CHANGE MASTER TO语句以启动复制过程：

```
shell> mysqldump --all-databases --master-data > dbdump.db
```

注意

如果您不使用--master-data，则需要手动将所有表锁定在单独的会话中。请参见第16.1.2.3节“获取复制主二进制日志坐标”。

可以使用mysqldump工具从转储中排除某些数据库。如果要选择要包含在转储中的数据库，请不要使用--all-databases。选择以下选项之一：

- 使用--ignore-table选项排除数据库中的所有表。
- 只使用--databases选项命名您想要转储的数据库。

有关更多信息，请参见第4.5.4节“mysqldump - 数据库备份程序”。

要导入数据，请将转储文件复制到从站，或者在远程连接到从站时从主站访问该文件。

16.1.2.4.2 使用原始数据文件创建数据快照

本节介绍如何使用组成数据库的原始文件创建数据快照。将这种方法与使用具有复杂缓存或日志记录算法的存储引擎的表配合使用需要额外的步骤来产生完美的“时间点”快照：初始复制命令可以省去缓存信息并记录更新，即使您已经获取全局读锁。存储引擎对此的响应取决于其崩溃恢复能力。

如果使用InnoDB表，则可以使用MySQL Enterprise Backup组件中的mysqlbackup命令来生成一致的快照。该命令记录从机上使用的快照对应的日志名称和偏移量。MySQL企业备份是作为MySQL企业订阅的一部分包含的商业产品。有关详细信息，请参见第29.2节“MySQL企业备份概述”。

如果主站和从站对ft_stopword_file, ft_min_word_len或ft_max_word_len具有不同的值，并且您正在复制具有全文索引的表，则此方法也无法可靠运行。

假设上述例外情况不适用于您的数据库，请使用冷备份技术来获取InnoDB表的可靠二进制快照：对MySQL服务器执行缓慢关闭，然后手动复制数据文件。

要在单个文件系统上存在MySQL数据文件时创建MyISAM表的原始数据快照，可以使用标准文件复制工具（例如cp或copy），远程复制工具（如scp或rsync），压缩工具（如zip或tar或文件系统快照工具（如dump））。如果您只复制特定数据库，则仅复制与这些表相关的文件。对于InnoDB，除非启用了innodb_file_per_table选项，否则所有数据库中的所有表都存储在系统表空间文件中。

以下文件不需要复制：

- 与mysql数据库相关的文件。
- 主信息库文件（如果使用的话）（请参见第16.2.4节“复制中继和状态日志”）。
- 主数据库的二进制日志文件。
- 任何中继日志文件。

根据您是否使用InnoDB表，请选择以下选项之一：

如果您正在使用InnoDB表，并且为了获得与原始数据快照最一致的结果，请在此过程中关闭主服务器，如下所示：

1. 获取一个读锁，并获得主服务器的状态。请参见第16.1.2.3节“获取复制主二进制日志坐标”。
2. 在单独的会话中，关闭主服务器：

```
shell> mysqladmin shutdown
```

3. 制作MySQL数据文件的副本。以下示例显示了执行此操作的常用方法。你只需要选择其中的一个：

```
shell> tar cf /tmp/db.tar ./data
shell> zip -r /tmp/db.zip ./data
shell> rsync --recursive ./data /tmp/dbdata
```

4. 重新启动主服务器。

如果您不使用InnoDB表，则可以从主服务器获取系统快照，而不必关闭服务器，如以下步骤中所述：

1. 获取一个读锁，并获得主服务器的状态。请参见第16.1.2.3节“获取复制主二进制日志坐标”。
2. 制作MySQL数据文件的副本。以下示例显示了执行此操作的常用方法。你只需要选择其中的一个：

```
shell> tar cf /tmp/db.tar ./data
shell> zip -r /tmp/db.zip ./data
shell> rsync --recursive ./data /tmp/dbdata
```

3. 在获取读锁的客户端，释放锁：

```
mysql> UNLOCK TABLES;
```

一旦创建了数据库的存档或副本，请在开始从属复制过程之前将这些文件复制到每个从属设备。