# MySQL服务器管理

MySQL服务器（**[mysqld](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html" \o "4.3.1 mysqld  -  MySQL服务器)**）是MySQL进行大部分工作的主要程序。本附录概述了MySQL Server，涵盖了一般的服务器管理，具体包括：

* 服务器配置，主要包括MySQL配置相关的系统变量、状态变量。
* 数据目录，特别是mysql 系统数据库。
* 服务器日志文件，包括错误日志、常规查询日志、二进制日志、慢查询日志、DDL日志（元数据日志）

中继日志。

* 在单台机器上管理多台服务器

有关管理主题的其他信息，请参阅：

* [安全](https://dev.mysql.com/doc/refman/5.7/en/security.html)
* [备份和恢复](https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html)
* [复制](https://dev.mysql.com/doc/refman/5.7/en/replication.html)

这三个都是非常专业的主题，本附录只是最为一般性的管理工作。

## 1、MySQL服务器

[**mysqld**](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html)是MySQL服务器。以下讨论涵盖了MySQL服务器配置相关的主题，主要包括：

* 服务器支持的启动选项。你可以通过配置文件或者在命令行中指定这些选项。
* 服务器系统变量。这些变量**反映了启动选项**的当前状态和值，其中一些可以在服务器运行时进行修改。
* 服务器状态变量。这些变量包含运行时MySQL服务器的统计信息。
* 设置服务器SQL模式。该设置会影响SQL语句的某些语义，以及为了兼容其他数据库的SQL语句而设置。
* 服务器关机过程。根据表的类型（事务性或非事务性）以及是否使用复制来考虑性能和可靠性。

关于在MySQL 的某个版本中添加的、不推荐使用的或者删除的MySQL服务器变量和选项的列表，请参见具体官方手册的说明，一般在第一章就会有。例如：对于MySQL 5.7就是在1.5节进行的说明。

**1.1 配置MySQL服务器**

MySQL服务器**[mysqld](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html" \o "4.3.1 mysqld  -  MySQL服务器)**具有许多命令选项和系统变量，可以在MySQL服务器启动时设置它们来配置其操作。要确定服务器使用的命令选项和系统变量的值，请执行以下命令（注意是在OS的终端shell下）：

shell> mysqld --verbose --help

该命令生成所有**[mysqld](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html" \o "4.3.1 mysqld  -  MySQL服务器)** 的选项和可配置的系统变量的列表。

查看当前服务器运行时的系统变量的值，请连接到服务器并执行如下语句：

mysql> SHOW VARIABLES;

查看运行的服务器的一些统计信息，请执行以下语句：

mysql> SHOW STATUS;

系统变量和状态信息也可以使用 [**mysqladmin**](https://dev.mysql.com/doc/refman/5.7/en/mysqladmin.html)命令查看：

shell> mysqladmin variables

shell> mysqladmin extended-status

MySQL使用非常灵活的算法，因此通常需要很少的内存就可以运行MySQL服务器。然而，由于MySQL是数据库服务器，缓存是非常重要的，因此给予MySQL更多的内存通常会提高MySQL的性能。调整MySQL服务器配置时，两个最重要的变量是[key\_buffer\_size](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html" \l "sysvar_key_buffer_size)和 [table\_open\_cache](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_table_open_cache)。在更改任何其他变量之前，你应该首先确信已经设置好了这些变量。下边举一些例子：

* 中等数量的客户端、拥有1-2GB的内存、有许多数据库表，要求具有高的性能，推荐使用以下配置：

shell> mysqld\_safe --key\_buffer\_size=384M --table\_open\_cache=4000 \ --sort\_buffer\_size=4M --read\_buffer\_size=1M &

* 只有256MB的内存，只有几个表，但你需要做很多排序，你可以如下配置：

shell> mysqld\_safe --key\_buffer\_size=64M --sort\_buffer\_size=1M

除非已将**[mysqld](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html" \o "4.3.1 mysqld  -  MySQL服务器)**配置为为每个连接使用非常少的内存，否则如果有非常多的并发连接，则可能会发生交换问题。 如果你有足够的内存用于所有连接，那么**[mysqld](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html" \o "4.3.1 mysqld  -  MySQL服务器)**的性能会更好。简单概括就是如果内存足够大，MySQL会使用大量内存，通过内存缓存，减少对磁盘的操作。进而提升性能。

如果在命令行中为**[mysqld](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html" \o "4.3.1 mysqld  -  MySQL服务器)**或**[mysqld\_safe](https://dev.mysql.com/doc/refman/5.7/en/mysqld-safe.html" \o "4.3.2 mysqld_safe  -  MySQL服务器启动脚本)**指定了一个选项 ，则它仅对该服务器有效。要在每次运行服务器时使用该选项，请将其放在配置文件中。

MySQL服务器具有许多可配的参数，你可以使用命令行选项或配置文件在服务器启动时进行更改。许多参数也可以在运行时更改。在MySQL 5.7.5之前，在Unix/Linux平台上， [**mysql\_install\_db**](https://dev.mysql.com/doc/refman/5.7/en/mysql-install-db.html)会创建一个默认的配置文件my.cnf，该文件存在于基本安装目录中。此文件是从分发包中的名为my-default.cnf的模板创建的。你可以在基本安装目录中找到该模板。当使用**[mysqld\_safe](https://dev.mysql.com/doc/refman/5.7/en/mysqld-safe.html" \o "4.3.2 mysqld_safe  -  MySQL服务器启动脚本)**启动时，服务器默认使用配置文件my.cnf。如果 my.cnf已经存在，那么 [**mysql\_install\_db**](https://dev.mysql.com/doc/refman/5.7/en/mysql-install-db.html)假定它正在被使用，这时， [**mysql\_install\_db**](https://dev.mysql.com/doc/refman/5.7/en/mysql-install-db.html)将创建一个名为my-new.cnf的新文件。**注意**：从MySQL 5.7.18起，my-default.cnf不再包含在分发包中或由分发包安装。除了一个特例（将[sql\_mode](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html" \l "sysvar_sql_mode)系统变量设置为NO\_ENGINE\_SUBSTITUTION，STRICT\_TRANS\_TABLES）外，默认配置文件中的设置均被注释了。关于MySQL都有哪些配置选项，系统变量、状态变量的信息，可以到MySQL的官方手册去查看。例如对于MySQL 5.7的手册，5.1.3节列出了所有信息，5.1.4节对配置选项进行了详细说明，5.1.5节对系统变量进行了详细说明。

由上边一系列的描述，我们知道MySQL服务器非常灵活，很多地方都是可配的，涉及到MySQL的方方面面，有的通过配置项来设置，有的是MySQL维护的系统变量，很多情况下，都是一个配置项对应一个系统变量，当然了，也有不对应的情况，有的只有配置项，有的只有系统变量。每个系统变量都有一个默认值。你可以使用命令行或者配置文件中的选项在服务器启动时设置系统变量。同时大多数系统变量可以在服务器运行时通过[SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法) 语句动态更改 ，而无需停止并重新启动服务器。你还可以在表达式中使用系统变量的值。

系统变量存在两个范围：全局变量--影响服务器的整体运行；会话变量--影响各个客户端连接的操作。给定的系统变量可以同时具有全局值和会话值。全局系统变量和会话系统变量的关系如下：

* 当服务器启动时，它会将每个全局变量初始化为其默认值。这些默认值可以通过在命令行或选项文件中指定的选项来更改。
* 服务器还为连接的每个客户端维护一组会话变量。客户端的会话变量在客户端连接服务器时使用相应的全局变量的当前值进行初始化。例如，客户机的SQL模式由会话系统变量[sql\_mode](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html" \l "sysvar_sql_mode)值控制，会话值在客户端连接到服务器时由对应的全局系统变量[sql\_mode](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html" \l "sysvar_sql_mode)值进行初始化。

对于某些系统变量，会话值不会从对应的全局值初始化，具体的参考MySQL手册的说明，对于MySQL5.7，5.1.5节对系统变量进行了详细说明。

通过使用命令行或配置文件中的选项可以在服务器启动时设置全局系统变量的值。当使用启动选项设置一个变量的时候，如果该选项需要一个数字值，那么可以给该值加一个后缀，K， M或G（大写或小写均可），以指示该值乘以1024，1024 **2**或1024 **3** ; 即，分别为千字节，兆字节或千兆字节的单位。因此，通过以下命令启动服务器，其查询高速缓存大小为16MB，最大数据包大小为1GB：

mysqld --query\_cache\_size=16M --max\_allowed\_packet=1G

也可以在配置文件中指定：

[mysqld]

query\_cache\_size=16M

max\_allowed\_packet=1G

要对[SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法) 语句在运行时可以设置的系统变量的最大值进行限制，请在服务器启动时以--maximum-***var\_name***=***value***格式来指定此最大值 。例如，为了防止在服务器运行时通过将query\_cache\_size增加到超过32MB，请用--maximum-query\_cache\_size=32M选项启动服务器。

许多系统变量是动态的，可以在运行时通过[SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法) 语句进行更改 。关于哪些变量是动态的，可以参考MySQL的官方手册，对于MySQL 5.7，请参见 [第5.1.6.2节“动态系统变量”](https://dev.mysql.com/doc/refman/5.7/en/dynamic-system-variables.html)。要通过[SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法)语句更改系统变量 ，请使用变量的名字，并在前面加上一个可选的修饰符：

* 要指示一个变量是全局变量，需要在其名称前面加上关键字GLOBAL或 @@global.限定词：

SET GLOBAL max\_connections = 1000;

SET @@global.max\_connections = 1000;

为了设置全局的系统变量，需要SUPER权限。

* 要指示一个变量是会话变量，需要在其名称前面加上关键字SESSION或者 @@session.或者@@ 限定符：

SET SESSION sql\_mode = 'TRADITIONAL';

SET @@session.sql\_mode = 'TRADITIONAL';

SET @@sql\_mode = 'TRADITIONAL';

设置会话变量通常不需要特殊权限，但是有些特例需要 SUPER特权（如 sql\_log\_bin）。客户端可以更改自己的会话变量，但不能改变任何其他客户端的会话变量。

* LOCAL和@@local.是SESSION和 @@session.的同义词
* 如果不存在修饰符，则 SET 默认更改会话变量。
* 在如下几种情况下会发生错误：
  + 在设置只有会话值的变量时 使用SET GLOBAL（或@@global.）
  + 设置仅具有全局值的变量时 省略GLOBAL（或 @@global.）：
  + 在设置只具有全局值的变量时 使用[SET SESSION](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法)（或@@SESSION.）

变量名称前面的修饰符仅适用于系统变量。尝试将其应用于用户定义的变量、存储过程或函数参数或存储过程的局部变量时也会出现错误。

一个[SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法) 语句可以为多个变量赋值，用逗号分隔。例如：以下语句将值分配给用户定义的变量和系统变量：

SET @x = 1, SESSION sql\_mode = '';

如果你设置了多个系统变量，则语句中的最后的 GLOBAL或SESSION修饰符将用于修饰后续未分配的变量，例如：

SET GLOBAL sort\_buffer\_size = 1000000, SESSION sort\_buffer\_size = 1000000;

SET @@global.sort\_buffer\_size = 1000000, @@local.sort\_buffer\_size = 1000000;

SET GLOBAL max\_connections = 1000, sort\_buffer\_size = 1000000;

如果SET语句中的任何变量赋值失败，则整个语句将失败，并且不会更改任何变量。

如果更改会话系统变量，则该值在会话中保持有效。更改对其他会话没有影响。

如果更改全局系统变量，则该值将被保存并用于新会话，直到将变量更改为其他值或服务器退出为止。访问全局变量的任何客户端都可以看到更改。但是，更改仅对更改后连接到服务器的客户端的会话变量产生影响。更改全局变量不会影响任何当前已连接客户端的会话变量（包括发出该SET语句的会话 ）。

要使全局系统变量永久性的改变，即使服务器重新启动也保持改变，你应该设置配置文件对应的选项。

要将GLOBAL值设置为编译的MySQL默认值，或将SESSION变量设置为当前对应的GLOBAL值，请将该变量设置为该值DEFAULT。例如，以下两个语句在将会话值设置[max\_join\_size](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html" \l "sysvar_max_join_size)为当前全局值时是相同的：

SET @@session.max\_join\_size=DEFAULT; SET @@session.max\_join\_size=@@global.max\_join\_size;

并非所有系统变量都可以设置为DEFAULT。在这种情况下，将DEFAULT结果分配给错误。

尝试分配DEFAULT 给用户定义的变量，存储过程或函数参数或存储的程序局部变量时发生错误。

要引用表达式中的系统变量的值，请使用@@-modifiers之一。例如，您可以在这样的[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html" \o "13.2.9选择语法) 语句中检索值：

SELECT @@global.sql\_mode, @@session.sql\_mode, @@sql\_mode;

对于将表达式中的系统变量引用 （而不是or或 ），MySQL返回会话值（如果存在），否则返回全局值。这不同于 ，它总是指会话值。@@***var\_name***@@global.@@session.SET @@***var\_name*** = ***expr***

在服务器启动时设置变量时，可以使用用于指定值乘数的后缀，但不能[SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法) 在运行时设置值。另一方面， [SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html)您可以使用表达式分配变量的值，当在服务器启动时设置变量时，该值不正确。例如，以下第一行在服务器启动时是合法的，但第二个不是：

shell> mysql --max\_allowed\_packet=16M shell> mysql --max\_allowed\_packet=16\*1024\*1024

相反，以下第二行在运行时是合法的，但第一个不是：

mysql> SET GLOBAL max\_allowed\_packet=16M; mysql> SET GLOBAL max\_allowed\_packet=16\*1024\*1024;

**注意**

可以通过将[SET](https://dev.mysql.com/doc/refman/5.7/en/set-variable.html" \o "13.7.4.1变量分配的SET语法) 语句设置为ON或 1或通过将其设置为OFF或禁用来禁用 某些系统变量 0。但是，要在命令行或选项文件中设置此变量，必须将其设置为1或0; 将其设置为ON或OFF 将不起作用。例如，在命令行上， [--delay\_key\_write=1](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_delay_key_write)但是 [--delay\_key\_write=ON](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_delay_key_write)没有。

要显示系统变量名和值，请使用 [SHOW VARIABLES](https://dev.mysql.com/doc/refman/5.7/en/show-variables.html)语句：

mysql> SHOW VARIABLES;

使用一个[LIKE](https://dev.mysql.com/doc/refman/5.7/en/string-comparison-functions.html" \l "operator_like)子句，语句仅显示与模式匹配的变量。要获取特定的变量名称，请使用[LIKE](https://dev.mysql.com/doc/refman/5.7/en/string-comparison-functions.html" \l "operator_like) 如下所示的子句：

SHOW VARIABLES LIKE 'max\_join\_size'; SHOW SESSION VARIABLES LIKE 'max\_join\_size';

要获取名称与模式匹配的变量列表，请%在[LIKE](https://dev.mysql.com/doc/refman/5.7/en/string-comparison-functions.html" \l "operator_like)子句中使用 通配符 ：

SHOW VARIABLES LIKE '%size%'; SHOW GLOBAL VARIABLES LIKE '%size%';

通配符可以在图案中的任何位置进行匹配。严格来说，因为\_是匹配任何单个字符的通配符，所以应该将其转义为\\_符合字符。在实践中，这是很少必要的。

因为[SHOW VARIABLES](https://dev.mysql.com/doc/refman/5.7/en/show-variables.html" \o "13.7.5.39 SHOW VARIABLES语法)，如果既不指定GLOBAL也不指定SESSION，MySQL返回SESSION值。

GLOBAL在设置时 要求关键字的原因 - GLOBAL只有变量，而不是检索它们是为了防止将来出现问题：

* 我们要删除与SESSION变量具有相同名称的GLOBAL变量，具有该[SUPER](https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html" \l "priv_super)权限的客户端可能会意外地更改GLOBAL 变量，而不仅仅是SESSION 自身连接的变量。
* 我们要添加与SESSION变量相同名称的GLOBAL变量，那么想要更改GLOBAL变量的客户端可能会发现只有自己的SESSION变量已更改。

4、MySQL服务器日志

对于服务端来说，日志是非常重要的东西，它可以帮助你定位服务器问题，了解服务器的活动，MySQL也由几种日志来帮助我们了解其活动，MySQL具有如下几种服务器日志：

| 日志类型 | 写入日志的信息 |
| --- | --- |
| 错误日志 | mysqld的启动、停止或运行中的问题 |
| 一般的查询日志 | 从客户端收到的客户端连接和语句 |
| 二进制日志 | 更改数据的语句（也用于复制） |
| 中继日志 | 从复制环境中主服务器收到的数据更改的日志 |
| 慢查询日志 | 执行时间超过long\_query\_time的查询（单位为妙） |
| DDL日志（元数据日志） | 由DDL语句执行的元数据操作 |