# 扩展MySQL

本章内容主要参考MySQL 5.7官方手册的第28章，里边会加入一些自己的理解。主要是站在源码的角度介绍如何去扩展MySQL，来加入自己的功能。

## 1 MySQL内幕

### 1.1 MySQL线程

MySQL服务器创建以下线程：

* 连接管理器线程处理服务器监听的网络接口上的客户端连接请求。在所有平台上，一个经理线程处理TCP / IP连接请求。在Unix上，这个管理器线程还处理Unix套接字文件连接请求。在Windows上，管理员线程处理共享内存连接请求，另一个处理命名管道连接请求。服务器不会创建线程来处理它听不到的接口。例如，启用了不支持命名管道连接的Windows服务器不会创建一个线程来处理它们。
* 连接管理器线程将每个客户端连接与专门用于处理该连接的身份验证和请求处理的线程相关联。经理线程在必要时创建一个新线程，但是先尝试避免这样做，首先查询线程缓存，以查看它是否包含可用于连接的线程。当连接结束时，如果缓存未满，则其线程返回到线程高速缓存。

有关调整控制线程资源的参数的信息，请参见[第8.12.5.1节“MySQL如何为客户端连接使用线程”](https://dev.mysql.com/doc/refman/5.7/en/connection-threads.html" \o "8.12.5.1 MySQL如何使用线程进行客户端连接)。

* 在主复制服务器上，来自从属服务器的连接将像客户端连接一样处理：每个连接的从站有一个线程。
* 在从属复制服务器上，启动I / O线程以连接主服务器并从中读取更新。开始SQL线程来应用从主机读取的更新。这两个线程独立运行，可以独立启动和停止。
* 信号线程处理所有信号。该线程还通常处理警报和呼叫， process\_alarm()以在长时间闲置的连接上强制超时。
* 如果InnoDB使用，默认情况下会有额外的读写线程。这些数目由受控 [innodb\_read\_io\_threads](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html" \l "sysvar_innodb_read_io_threads)和[innodb\_write\_io\_threads](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html" \l "sysvar_innodb_write_io_threads) 参数。请参见[第14.14节“InnoDB启动选项和系统变量”](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html" \o "14.14 InnoDB启动选项和系统变量)。
* 如果使用该 选项启动服务器 ，则会创建专用线程以每秒钟刷新所有表。 [--flush\_time=](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html" \l "sysvar_flush_time)***[val](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html" \l "sysvar_flush_time)val***
* 如果事件调度程序处于活动状态，则调度程序有一个线程，当前运行的每个事件都有一个线程。请参见[第23.4.1节“事件计划程序概述”](https://dev.mysql.com/doc/refman/5.7/en/events-overview.html" \o "23.4.1事件计划程序概述)。

**[mysqladmin进程列表](https://dev.mysql.com/doc/refman/5.7/en/mysqladmin.html" \o "4.5.2 mysqladmin  - 用于管理MySQL服务器的客户端)**仅显示连接，复制和事件线程。

### 1.2 MySQL测试套件

包含在Unix源代码和二进制发行版中的测试系统使用户和开发人员可以对MySQL代码执行回归测试。这些测试可以在Unix上运行。

您也可以编写自己的测试用例。有关MySQL测试框架的信息（包括系统要求），请参阅[http://dev.mysql.com/doc/mysqltest/2.0/en/上](https://dev.mysql.com/doc/mysqltest/2.0/en/" \t "_top)提供的手册。

目前的测试用例集不会测试MySQL中的所有内容，但它应该捕获SQL处理代码，操作系统或库问题中最明显的错误，并且在测试复制时是非常彻底的。我们的目标是让测试覆盖100％的代码。我们欢迎对我们的测试套件的贡献。您可能特别想要提供检查系统关键功能的测试，因为这样可以确保所有未来的MySQL版本都能够与您的应用程序配合使用。

测试系统包括测试语言解释器（**mysqltest**），运行所有测试（**mysql-test-run.pl**）的Perl脚本，以特殊测试语言编写的实际测试用例及其预期结果。要在构建之后在系统上运行测试套件，请从源根目录键入 **make test**，或将位置更改为mysql-test目录并键入**./mysql-test-run.pl**。如果您已经安装了二进制分发，请将位置更改 mysql-test为安装根目录下的目录（例如/usr/local/mysql/mysql-test），然后运行 **./mysql-test-run.pl**。所有测试都应该成功。如果没有，请随时尝试找出为什么并报告问题，如果它指示MySQL中的错误。请参见 [第1.7节“如何报告错误或问题”](https://dev.mysql.com/doc/refman/5.7/en/bug-reports.html" \o "1.7如何报告错误或问题)。

如果一个测试失败，您应该运行 **mysql-test-run.pl**与 --force选项来检查是否有任何其他测试失败。

如果您有要运行测试套件的机器上运行的**[mysqld](https://dev.mysql.com/doc/refman/5.7/en/mysqld.html" \o "4.3.1 mysqld  -  MySQL服务器)**的副本，则不必停止它，只要它不使用端口 9306或9307。如果这些端口中的任何一个被采取，您应该将 MTR\_BUILD\_THREAD环境变量设置为适当的值，并且测试套件将为主器件，从器件和NDB使用一组不同的端口）。例如：

shell> export MTR\_BUILD\_THREAD=31 shell> ./mysql-test-run.pl [*options*] [*test\_name*]

在mysql-test目录中，您可以使用**./mysql-test-run.pl *test\_name***运行单个测试用例。

如果您有关于测试套件的问题，或者有一个测试用例来贡献，请发送电子邮件到MySQL internals邮件列表。请参见 [第1.6.2节“MySQL邮件列表”](https://dev.mysql.com/doc/refman/5.7/en/mailing-lists.html" \o "1.6.2 MySQL邮件列表)。

## 2、MySQL插件API

作者：许富博

版权所有，文章以学习和交流为主，切勿用于商业用途。

限于本人水平有限，欢迎大家随时指正，联系方式：

[xufubobo@gmail.com](mailto:xufubobo@gmail.com)

[xufubobo@163.com](mailto:xufubobo@163.com)

[1332841493@qq.com](mailto:1332841493@qq.com)