1. MYSQL介绍
2. 架构

MYSQL采用C/S架构，服务器程序直接与存储的数据交互，多个客户端程序可以连接到这个服务器程序，向服务器发送增删改查的请求，然后服务器程序对数据进行处理。

1. 服务器程序的启动
2. 类UNIX系统
3. mysqld

mysqld可执行文件表示MYSQL的服务器程序，运行这个可执行文件就可以启动一个MYSQL服务器进程，不常用。

1. mysqld\_safe

一个启动脚本，间接地调用mysqld并持续监控服务器的运行状态，将出错信息和其他诊断信息输出到错误日志中。

1. mysqld.server

一个启动脚本，会间接地调用mysqld\_safe，在后面添加start参数就可以启动服务器程序：

mysqld.server start

该文件其实是一个链接文件，对应的实际文件是 ../supportfiles/mysql.server。也可以使用该脚本来关闭服务器程序：

mysqld.server stop

1. mysqld\_multi

运行多个服务器实例，启动或停止多个服务器进程，同时报告它们的运行状态。

1. Windows系统
2. 手动启动

直接单击mysqld可执行文件。

1. 服务方式启动

如果要长时间运行某程序，就可以将它注册为一个Windows服务，由操作系统来管理。注册服务如下所示：

“完整可执行文件路径” –install [-manual] [服务名]

-manual 表示手动启动该服务，默认的服务名是MYSQL。接下来就可以启动MYSQL服务器服务了：

net start MYSQL

关闭服务：

net stop MYSQL

1. 客户端程序的启动

通过mysql可执行文件，客户端可以与服务器程序进行交互，不过需要通过传递一些参数来，如-h 服务器程序的IP，-u登录的用户名，-p该用户的密码，-P服务器程序监听的端口号，注意密码与p之间是没有空格的，不然会报错。

1. 客户端与服务器的连接

MYSQL客户端程序与服务端程序的通信实质上是一个进程间通信的过程。

1. TCP/IP

客户端和服务器可能在不同的主机中，因此必须通过网络进行通信，MySQL采用TCP作为网络通信协议，服务器默认监听3306端口号，端口的范围为 0 – 65535。

1. 命名管道和共享内存

Windows用户可以使用命名管道和共享内存进行进程间的通信。

命名管道启动：在启动服务器程序的命令中加上 –enablee-named-pipe参数，在启动客户端中加上 –pipe参数。

共享内存启动：服务器程序启动加上—shared-memory参数，客户端程序启动加上—protocol=memory参数。

1. UNIX域套接字

C/S都运行在类Unix系统中，可使用UNIX域套接字进行通信。启动客户端时指定—protocol=socket参数，服务器程序默认监听的Unix域套接字文件为/tmp/mysql.sock，客户端程序也默认连接到这个Unix域套接字文件。

1. MySQL处理请求流程
2. 概述

服务器程序在处理来自客户端的查询请求时，分为三步：连接管理，解析与优化，存储引擎。

1. 连接管理

每当有一个客户端进程连接到服务器进程时，服务器进程都会创建一个线程专门处理与这个客户端的交互，当该客户端退出时会与服务器断开连接，而服务器会将该线程缓存起来，等待另一个新的客户端的连接。客户端发送的连接请求是需要验证的，如果认证失败会拒绝连接。如果C/S运行在不同机器上，会采用传输层安全性（TLS）协议对连接进行加密。

1. 解析与优化
2. 查询缓存

MySQL服务器程序处理请求时会首先查询之前的缓存，如果有相同的请求就会直接从缓存中返回，这个查询缓存可以在不同的客户端之间共享。如果查询请求有任何字符上的不同，都会导致缓存不命中，且如果查询请求中包含有某些系统函数，用户自定义变量和函数，系统表，则这个请求不会被缓存。如系统函数NOW查询最新的当前时间，该函数两次调用肯定是不同的结果，因此不会被缓存。

MySQL的缓存系统会监控涉及的每张表，只要该表的结构或数据被修改，则所有相关的查询缓存都会变为无效并从查询缓冲中删除。虽然查询缓存能够提高系统性能，但是为了维护这块缓存也会有一些开销，从MySQL5.7.20开始不推荐使用，到了8.0直接删除了。

1. 语法解析

查询缓存没有命中，就会进入正式的查询中，客户端发送过来一个SQL语句，首先要对这个语句进行分析，判断语法是否正确，然后从文本中将要查询的表，各种查询条件都提取出来放到MySQL服务器内部使用的一些数据结构上。

这个过程类似编译器的 词法解析->语法分析->语义分析。

1. 查询优化

语法解析后，服务器程序获得了需要的信息，但是为了使SQL语句查询效率更高，需要优化SQL语句，如将外连接转换为内连接，表达式简化等等。优化的结果是生成一个执行计划，这个执行计划表明了应该使用哪些索引执行查询，以及表之间的连接顺序是什么样的。我们可以使用EXPLAIN语句来查看某个语句的执行计划。

1. 存储引擎

MySQL服务器将数据的存储和提取操作都封装到了存储引擎中，表是由一行一行的记录组成的，这只是逻辑概念，存储引擎负责真正数据在物理上的存储，读取。不同的存储引擎负责不同的功能，不同存储引擎管理的表可能有不同的存储结构，存取算法。

为了方便管理，MySQL服务器处理请求的过程被划分为了server层和存储引擎层，各种不同的存储请求为server层提供统一的调用接口，其中包含了不同用途的底层函数，如读取索引第一条记录等等。

server层在判断某条记录符合要求之后，首先将其发送到一个缓冲区中，待到该缓冲区满了，才会向客户端发送真正的记录。

1. 常用的存储引擎

MySQL在5.5.5之前默认的存储引擎为MyISAM，之后改为了InnoDB，InnoDB支持事务，分布式事务以及事务的部分回滚，这些大部分引擎都没有。

查询当前服务器程序支持的存储引擎：

SHOW ENGINES;

我们可以为不同的表设置不同的存储引擎，不同的表可以有不同的物理存储结构，不同的读取和写入方式。在建表时直接指定引擎：

CREATE TABLE(

…

)ENGINE = …;

修改表的存储引擎：

ALTER TABLE 表名 ENGINE=…;