**第一种：命令行修改**。

>mysql -uuser -ppassword(命令行登录MySQL)

mysql>show variables like 'max\_connections';(查可以看当前的最大连接数)

msyql>set global max\_connections=1000;(设置最大连接数为1000，可以再次查看是否设置成功)

mysql>exit(推出)

这种方式有个问题，就是设置的最大连接数只在mysql当前服务进程有效，一旦mysql重启，又会恢复到初始状态。因为mysql启动后的初始化工作是从其配置文件中读取数据的，而这种方式没有对其配置文件做更改。

**第二种：修改配置文件。**

这 种方式说来很简单，只要修改MySQL配置文件my.ini 或 my.cnf的参数max\_connections，将其改为max\_connections=1000，然后重启MySQL即可。但是有一点最难的就是my.ini这个文件在哪找。通常有两种可能，一个是在安装目录下（这是比较理想的情况），另一种是在数据文件的目录下，安装的时候如果没有人为改变目录的话，一般就在C:/ProgramData/MySQL往下的目录下。

[查看mysql数据库连接数、并发数相关信息](http://blog.chinaunix.net/uid-20382003-id-3022768.html)

**连接数：**

就好比你用navicat去连mysql，同时支持600个用navicat去连的用户，超过600个的下一个就需要等待了，这里并不是说只能保持600个用户同时访问，因为你数据库和你应用之间肯定隔着中间件，而中间件中的连接池会帮你管理好每个连接，一般如果代码写的好事物控制得当，一个事物完成连接会被连接池收回重复利用，所以不是说一个用户登录你的系统或网站就一直占用着。：

**IOPS (Input/Output Operations Per Second)**：

即每秒进行读写（I/O）操作的次数，多用于数据库等场合，衡量随机访问的性能。存储端的IOPS性能和主机端的IO是不同的，IOPS是指存储每秒可接受多少次主机发出的访问，主机的一次IO需要多次访问存储才可以完成。例如，主机写入一个最小的数据块，也要经过“发送写入请求、写入数据、收到写入确认”等三个步骤，也就是3个存储端访问。一个连接过来，可能要SELECT不止一次，每次也不止一次的IO

**1.mysql> show status like 'Threads%';**  
+-------------------+-------+  
| Variable\_name     | Value |  
+-------------------+-------+  
| Threads\_cached    | 58    |  
| Threads\_connected | 57    |   ###这个数值指的是打开的连接数  
| Threads\_created   | 3676  |  
| Threads\_running   | 4     |   ###这个数值指的是激活的连接数，这个数值一般远低于connected数值

+-------------------+-------+

Threads\_connected 跟**show processlist**结果相同，表示当前连接数。准确的来说，Threads\_running是代表当前并发数

这是是查询数据库当前设置的最大连接数

**2.mysql> show variables like '%max\_connections%';**  
+-----------------+-------+  
| Variable\_name   | Value |  
+-----------------+-------+  
| max\_connections | 1000  |  
+-----------------+-------+

可以在/etc/my.cnf里面设置数据库的最大连接数

[mysqld]

max\_connections = 1000  
  
max\_connections 参数可以用于控制数据库的最大连接数：  
**3.mysql> show variables like '%connect%';**  
+--------------------------+-------------------+  
| Variable\_name            | Value             |  
+--------------------------+-------------------+  
| character\_set\_connection | latin1            |  
| collation\_connection     | latin1\_swedish\_ci |  
| connect\_timeout          | 10                |  
| init\_connect             |                   |  
| max\_connect\_errors       | 10                |  
| max\_connections          | 4000              |  
| max\_user\_connections     | 0                 |  
+--------------------------+-------------------+

很多开发人员都会遇见”MySQL: ERROR 1040: Too many connections”的异常情况，造成这种情况的一种原因是访问量过高，MySQL服务器抗不住，这个时候就要考虑增加从服务器分散读压力；另一种原因就是**MySQL配置文件中max\_connections值过小。**

首先，我们来查看mysql的最大连接数：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | mysql> show variables like '%max\_connections%';  +-----------------+-------+  | Variable\_name  | Value |  +-----------------+-------+  | max\_connections | 151  |  +-----------------+-------+  1 row in set (0.00 sec) |

其次，查看服务器响应的最大连接数:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | mysql> show global status like 'Max\_used\_connections';  +----------------------+-------+  | Variable\_name    | Value |  +----------------------+-------+  | Max\_used\_connections | 2   |  +----------------------+-------+  1 row in set (0.00 sec) |

可以看到服务器响应的最大连接数为2，远远低于mysql服务器允许的最大连接数值。

show status like '%max\_connections%'; ##mysql最大连接数  
set global max\_connections=1000 ##重新设置  
show variables like '%max\_connections%'; ##查询数据库当前设置的最大连接数

show global status like 'Max\_used\_connections'; ##服务器响应的最大连接数

show status like 'Threads%';

Variable\_name Value  
Threads\_cached 0 ##mysql管理的线程池中还有多少可以被复用的资源  
Threads\_connected 152 ##打开的连接数  
Threads\_created 550 ##表示创建过的线程数，如果发现Threads\_created值过大的话，表明MySQL服务器一直在创建线程，这也是比较耗资源，可以适当增加配置文件中thread\_cache\_size值，查询服务器  
Threads\_running 1 ##激活的连接数，这个数值一般远低于connected数值，准确的来说，Threads\_running是代表当前并发数