[MySQL server has gone away 问题的解决方法](https://www.cnblogs.com/fnlingnzb-learner/p/5984795.html)

mysql出现ERROR : (2006, 'MySQL server has gone away') 的问题意思就是指client和MySQL server之间的链接断开了。

造成这样的原因一般是sql操作的时间过长，或者是传送的数据太大(例如使用insert ... values的语句过长， 这种情况可以通过修改max\_allowed\_packed的配置参数来避免，也可以在程序中将数据分批插入)。

产生这个问题的原因有很多，总结下网上的分析：

**原因一. MySQL 服务宕了**

判断是否属于这个原因的方法很简单，进入mysql控制台，查看mysql的运行时长

mysql> show global status like 'uptime';  
+---------------+---------+  
| Variable\_name | Value   |  
+---------------+---------+  
| Uptime        | 3414707 |  
+---------------+---------+

1 row in set或者查看MySQL的报错日志，看看有没有重启的信息

如果uptime数值很大，表明mysql服务运行了很久了。说明最近服务没有重启过。  
如果日志没有相关信息，也表名mysql服务最近没有重启过，可以继续检查下面几项内容。

**原因二. mysql连接超时**

即某个mysql长连接很久没有新的请求发起，达到了server端的timeout，被server强行关闭。  
此后再通过这个connection发起查询的时候，就会报错server has gone away  
（大部分PHP脚本就是属于此类）

mysql> show global variables like '%timeout';  
+----------------------------+----------+  
| Variable\_name              | Value    |  
+----------------------------+----------+  
| connect\_timeout            | 10       |  
| delayed\_insert\_timeout     | 300      |  
| innodb\_lock\_wait\_timeout   | 50       |  
| innodb\_rollback\_on\_timeout | OFF      |  
| interactive\_timeout        | 28800    |  
| lock\_wait\_timeout          | 31536000 |  
| net\_read\_timeout           | 30       |  
| net\_write\_timeout          | 60       |  
| slave\_net\_timeout          | 3600     |  
| wait\_timeout               | 28800    |  
+----------------------------+----------+  
10 rows in set

wait\_timeout 是28800秒，即mysql链接在无操作28800秒后被自动关闭

**原因三. mysql请求链接进程被主动kill**

这种情况和原因二相似，只是一个是人为一个是MYSQL自己的动作

mysql> show global status like 'com\_kill';  
+---------------+-------+  
| Variable\_name | Value |  
+---------------+-------+  
| Com\_kill      | 21    |  
+---------------+-------+  
1 row in set原因四. Your SQL statement was too large.

当查询的结果集超过 max\_allowed\_packet 也会出现这样的报错。定位方法是打出相关报错的语句。

用select \* into outfile 的方式导出到文件，查看文件大小是否超过 max\_allowed\_packet ，如果超过则需要调整参数，或者优化语句。

mysql> show global variables like 'max\_allowed\_packet';  
+--------------------+---------+  
| Variable\_name      | Value   |  
+--------------------+---------+  
| max\_allowed\_packet | 1048576 |  
+--------------------+---------+  
1 row in set (0.00 sec)  
  
修改参数：

mysql> set global max\_allowed\_packet=1024\*1024\*16;  
mysql> show global variables like 'max\_allowed\_packet';  
+--------------------+----------+  
| Variable\_name      | Value    |  
+--------------------+----------+  
| max\_allowed\_packet | 16777216 |  
+--------------------+----------+  
1 row in set (0.00 sec)  
  
**以下是补充：**  
应用程序长时间的执行批量的MYSQL语句。执行一个SQL，但SQL语句过大或者语句中含有BLOB或者longblob字段。比如，图片数据的处理。都容易引起MySQL server has gone away。   
  
今天遇到类似的情景，MySQL只是冷冷的说：MySQL server has gone away。   
  
大概浏览了一下，主要可能是因为以下几种原因：   
一种可能是发送的SQL语句太长，以致超过了max\_allowed\_packet的大小，如果是这种原因，你只要修改my.cnf，加大max\_allowed\_packet的值即可。   
  
还有一种可能是因为某些原因导致超时，比如说程序中获取数据库连接时采用了Singleton的做法，虽然多次连接数据库，但其实使用的都是同一个连接，而且程序中某两次操作数据库的间隔时间超过了wait\_timeout（SHOW STATUS能看到此设置），那么就可能出现问题。最简单的处理方式就是把wait\_timeout改大，当然你也可以在程序里时不时顺手mysql\_ping()一下，这样MySQL就知道它不是一个人在战斗。   
  
**解决MySQL server has gone away**   
  
**1、应用程序长时间的执行批量的MYSQL语句。最常见的就是采集或者新旧数据转化。或者长时间闲置数据库连接（我的项目就是这样）**   
解决方案：   
在my.cnf文件中添加或者修改以下两个变量：

1 wait\_timeout=2880000

2 interactive\_timeout = 2880000

关于两个变量的具体说明可以google或者看官方手册。如果不能修改my.cnf，则可以在连接数据库的时候设置CLIENT\_INTERACTIVE，比如：

1 sql = "set interactive\_timeout=24\*3600";

2 mysql\_real\_query(...)

2、执行一个SQL，但SQL语句过大或者语句中含有BLOB或者longblob字段。比如，图片数据的处理   
解决方案：   
在my.cnf文件中添加或者修改以下变量：

1 max\_allowed\_packet = 10M(也可以设置自己需要的大小)

max\_allowed\_packet 参数的作用是，用来控制其通信缓冲区的最大长度。

最近在做一个项目，需要程序24小时开着，而中间会有很多闲置时间，于是每天早上过来第一次操作数据库时，就出现了“MySQL server has gone away”这样的错误提示，而这个问题的原因是由于数据库连接由于长时间没有操作而会被自动关闭。解决这个问题，我的经验有以下两点，或许对大家有用处：   
**第 一种方法：**当然是增加你的 wait-timeout值，这个参数是在my.cnf(在Windows下台下面是my.ini）中设置，我的数据库负荷稍微大一点，所以，我设置的值 为10，（这个值的单位是秒，意思是当一个数据库连接在10秒钟内没有任何操作的话，就会强行关闭，我使用的不是永久链接 （mysql\_pconnect),用的是mysql\_connect,关于这个wait-timeout的效果你可以在MySQL的进程列表中看到 （show processlist) ），你可以把这个wait-timeout设置成更大，比如300秒，呵呵，一般来讲300秒足够用了，其实你也可以不用设置，MySQL默认是8个小 时。情况由你的服务器和站点来定。   
**第二种方法：**   
这也是我个人认为最好的方法，即检查 MySQL的链接状态，使其重新链接。 （用mysql\_ping()）  
可能大家都知道有mysql\_ping这么一个函数，在很多资料中都说这个mysql\_ping的 API会检查数据库是否链接，如果是断开的话会尝试重新连接，但在我的测试过程中发现事实并不是这样子的，是有条件的，必须要通过 mysql\_options这个C API传递相关参数，让MYSQL有断开自动链接的选项（MySQL默认为不自动连接），但我测试中发现PHP的MySQL的API中并不带这个函数，你重新编辑MySQL吧，呵呵。但mysql\_ping这个函数还是终于能用得上的，只是要在其中有一个小小的操作技巧：

[复制代码](javascript:void(0);)

1 //使用mysql\_ping来自动检查重连。用到两个函数，一个是mysql\_ping，另外一个是mysql\_options。具体使用方法是在mysql\_real\_connect之前，mysql\_init之后，使用mysql\_options。用法如下：

2 char value = 1;

3 (void) mysql\_init (&mysql);

4 mysql\_options(&mysql, MYSQL\_OPT\_RECONNECT, (char \*)&value); //设置自动连接

5 //然后在以后mysql\_query之前首先使用mysql\_ping进行判断，如果连接已经断开，会自动重连。

6 mysql\_ping(&mysql);

7 //ping()这个函数先检测数据连接是否正常，如果被关闭，整个把当前脚本的MYSQL实例关闭，再重新连接。

8 //经过这样处理后，可以非常有效的解决MySQL server has gone away这样的问题，而且不会对系统造成额外的开销。

9 //mysql\_ping会改变mysql\_affected\_rows的返回值。所以最好是给该MYSQL句柄再加一个mutex（最好是读写锁）。当其它线程准备执行query的时候，就获取锁，执行完就释放。而这个执行mysql\_ping的线程在执行ping之间先尝试获取锁，如果获取失败，

10 //则继续sleep，放弃这一轮的ping。

[复制代码](javascript:void(0);)

附录：

**mysqli\_options()**

**定义和语法**

mysqli\_options() 函数设置额外的连接选项，用于影响连接行为。

mysqli\_options() 函数可以被调用若干次来设置若干个选项。

**注释：**mysqli\_options() 函数可以在 mysqli\_init() 之后和 mysqli\_real\_connect() 之前被调用。

**语法：**

1 mysqli\_options(connection,option,value);

connection：必需。规定要使用的 MySQL 连接。

option:必需。规定要设置的选项。可以是下列值中的一个：

* MYSQLI\_OPT\_CONNECT\_TIMEOUT - 以秒为单位的连接超时时间
* MYSQLI\_OPT\_LOCAL\_INFILE - 启用/禁用 LOAD LOCAL INFILE
* MYSQLI\_INIT\_COMMAND - 在连接到 MySQL 服务器之后的执行命令
* MYSQLI\_READ\_DEFAULT\_FILE - 从已命名的文件而不是 my.cnf 中读取选项
* MYSQLI\_READ\_DEFAULT\_GROUP - 从 my.cnf 或者 MYSQLI\_READ\_DEFAULT\_FILE 中指定的文件中的已命名组中读取选项
* MYSQLI\_SERVER\_PUBLIC\_KEY - 基于 SHA-256 认证的 RSA 公共密钥文件

value:必需。规定 *option* 的值。

**mysql\_ping()**

mysql\_ping指：本函数可用于空闲很久的脚本来检查服务器是否关闭了连接。

**定义和语法**

nt STDCALL mysql\_ping(MYSQL \*mysql);

描述：

检查与服务端的连接是否正常。连接断开时，如果自动重新连接功能未被禁用，则尝试重新连接服务器。该函数可被客户端用来检测闲置许久以后，与服务端的连接是否关闭，如有需要，则重新连接。

返回值：  
　　连接正常，返回0；如有错误发生，则返回非0值。返回非0值并不意味着服务器本身关闭掉，也有可能是网络原因导致网络不通。

错误码：  
CR\_COMMANDS\_OUT\_OF\_SYNC 命令以不正确的顺序执行

CR\_SERVER\_GONE\_ERROR 服务器连接断开

**。。。**

**语法**

mysql\_ping(connection)

connection：可选。规定 MySQL 连接。如果未规定，则使用上一个连接。