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（比如PHP）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（比如PHP）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。   
解决方案：   
在my.cnf文件中添加或者修改以下两个变量：   
wait\_timeout=2880000   
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：   
sql = "set interactive\_timeout=24\*3600";   
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：   
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。   
  
  
最近做网站有一个站要用到WEB网页采集器功能，当一个PHP脚本在请求URL的时候，可能这个被请求的网页非常慢慢，超过了mysql的 wait-timeout时间，然后当网页内容被抓回来后，准备插入到MySQL的时候，发现MySQL的连接超时关闭了，于是就出现了“MySQL server has gone away”这样的错误提示，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。   
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：   
这是我的的数据库操 作类中间的一个函数

复制代码 代码如下:

function ping(){   
if(!mysql\_ping($this->link)){   
mysql\_close($this->link); //注意：一定要先执行数据库关闭，这是关键   
$this->connect();   
}   
}

我需要调用这个函数的代码可能是这样子的

复制代码 代码如下:

$str = file\_get\_contents('http://www.jb51.net');   
$db->ping();//经过前面的网页抓取后，或者会导致数据库连接关闭,检查并重新连接   
$db->query('select \* from table');

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。   
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。   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
　　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（比如PHP）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。   
　　解决方案：   
　　在my.cnf文件中添加或者修改以下两个变量：   
wait\_timeout=2880000   
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：   
sql = "set interactive\_timeout=24\*3600";   
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：   
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。  
**1、应用程序（比如PHP）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。**

**解决方案：**  
  
在my.ini文件中添加或者修改以下两个变量：  
wait\_timeout=**2880000**  
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：  
sql = "set interactive\_timeout=24\*3600";  
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：  
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。

**修改方法**

1） 方法1  
可以编辑my.cnf来修改（windows下my.ini）,在[mysqld]段或者mysql的server配置段进行修改。

max\_allowed\_packet = 20M  
如果找不到my.cnf可以通过

mysql --help | grep my.cnf  
去寻找my.cnf文件。

**2） 方法2**  
（很妥协，很纠结的办法）

进入mysql server

在mysql 命令行中运行

set global max\_allowed\_packet = 2\*1024\*1024\*10  
然后关闭掉这此mysql server链接，再进入。

show VARIABLES like '%max\_allowed\_packet%';  
查看下max\_allowed\_packet是否编辑成功  
  
－－－－－－－－－－－－ **以下是网络搜索的资料** －－－－－－－－－－－－－－－－－－－

也许其他人遇到这个问题，不一定是这儿的原因，那么，就把我在网上找到比较全面的分析放到下面：  
有两篇，第一篇比较直观，第二篇比较深奥。  
解决MySQL server has gone away 2009-01-09 16:23:22  
  
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（比如PHP）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。  
  
解决方案：  
  
在my.cnf文件中添加或者修改以下两个变量：  
  
wait\_timeout=2880000  
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：  
  
sql = "set interactive\_timeout=24\*3600";  
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：  
  
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  
  
MySQL: 诡异的MySQL server has gone away及其解决  
  
在Mysql执行show status，通常更关注缓存效果、进程数等，往往忽略了两个值：  
  
Variable\_name Value   
Aborted\_clients 3792   
Aborted\_connects 376  
  
  
通常只占query的0.0x%，所以并不为人所重视。而且在传统Web应用上，query错误对用户而言影响并不大，只是重新刷新一下页面就OK了。最近的基础改造中，把很多应用作为service运行，无法提示用户重新刷新，这种情况下，可能就会影响到服务的品质。  
  
通过程序脚本的日志跟踪，主要报错信息为“MySQL server has gone away”。官方的解释是：  
  
The most common reason for the MySQL server has gone away error is that the server timed out and closed the connection.  
  
Some other common reasons for the MySQL server has gone away error are:  
  
You (or the db administrator) has killed the running thread with a KILL statement or a mysqladmin kill command.  
  
You tried to run a query after closing the connection to the server. This indicates a logic error in the application that should be corrected.  
  
A client application running on a different host does not have the necessary privileges to connect to the MySQL server from that host.  
  
You got a timeout from the TCP/IP connection on the client side. This may happen if you have been using the commands: mysql\_options(..., MYSQL\_OPT\_READ\_TIMEOUT,...) or mysql\_options(..., MYSQL\_OPT\_WRITE\_TIMEOUT,...). In this case increasing the timeout may help solve the problem.  
  
You have encountered a timeout on the server side and the automatic reconnection in the client is disabled (the reconnect flag in the MYSQL structure is equal to 0).  
  
You are using a Windows client and the server had dropped the connection (probably because wait\_timeout expired) before the command was issued.  
  
The problem on Windows is that in some cases MySQL doesn't get an error from the OS when writing to the TCP/IP connection to the server, but instead gets the error when trying to read the answer from the connection.  
  
In this case, even if the reconnect flag in the MYSQL structure is equal to 1, MySQL does not automatically reconnect and re-issue the query as it doesn't know if the server did get the original query or not.  
  
The solution to this is to either do a mysql\_ping on the connection if there has been a long time since the last query (this is what MyODBC does) or set wait\_timeout on the mysqld server so high that it in practice never times out.  
  
You can also get these errors if you send a query to the server that is incorrect or too large. If mysqld receives a packet that is too large or out of order, it assumes that something has gone wrong with the client and closes the connection. If you need big queries (for example, if you are working with big BLOB columns), you can increase the query limit by setting the server's max\_allowed\_packet variable, which has a default value of 1MB. You may also need to increase the maximum packet size on the client end. More information on setting the packet size is given in Section A.1.2.9, “Packet too large”.  
  
An INSERT or REPLACE statement that inserts a great many rows can also cause these sorts of errors. Either one of these statements sends a single request to the server irrespective of the number of rows to be inserted; thus, you can often avoid the error by reducing the number of rows sent per INSERT or REPLACE.  
  
You also get a lost connection if you are sending a packet 16MB or larger if your client is older than 4.0.8 and your server is 4.0.8 and above, or the other way around.  
  
It is also possible to see this error if hostname lookups fail (for example, if the DNS server on which your server or network relies goes down). This is because MySQL is dependent on the host system for name resolution, but has no way of knowing whether it is working — from MySQL's point of view the problem is indistinguishable from any other network timeout.  
  
You may also see the MySQL server has gone away error if MySQL is started with the --skip-networking option.  
  
Another networking issue that can cause this error occurs if the MySQL port (default 3306) is blocked by your firewall, thus preventing any connections at all to the MySQL server.  
  
You can also encounter this error with applications that fork child processes, all of which try to use the same connection to the MySQL server. This can be avoided by using a separate connection for each child process.  
  
You have encountered a bug where the server died while executing the query.  
  
  
  
据此分析，可能原因有3：  
  
1，Mysql服务端与客户端版本不匹配。  
  
2，Mysql服务端配置有缺陷或者优化不足  
  
3，需要改进程序脚本  
  
通过更换多个服务端与客户端版本，发现只能部分减少报错，并不能完全解决。排除1。  
  
对服务端进行了彻底的优化，也未能达到理想效果。在timeout的取值设置上，从经验值的10，到PHP默认的60，进行了多次尝试。而Mysql官方默认值(8小时)明显是不可能的。从而对2也进行了排除。(更多优化的经验分享，将在以后整理提供)  
  
针对3对程序代码进行分析，发现程序中大量应用了类似如下的代码(为便于理解，用原始api描述)：  
  
$conn=mysql\_connect( ... ... );  
  
... ... ... ...  
  
if(!$conn){ //reconnect  
  
$conn=mysql\_connect( ... ... );  
  
}  
  
mysql\_query($sql, $conn);  
  
这段代码的含义，与Mysql官方建议的方法思路相符[ If you have a script, you just have to issue the query again for the client to do an automatic reconnection. ]。在实际分析中发现，if(!$conn)并不是可靠的，程序通过了if(!$conn)的检验后，仍然会返回上述错误。  
  
对程序进行了改写：  
  
if(!conn){ // connect ...}  
  
elseif(!mysql\_ping($conn)){ // reconnect ... }  
  
mysql\_query($sql, $conn);  
  
经实际观测，MySQL server has gone away的报错基本解决。  
  
BTW： 附带一个关于 reconnect 的疑问，  
  
在php4x+client3x+mysql4x的旧环境下，reconnet的代码：  
  
$conn=mysql\_connect(...) 可以正常工作。  
  
但是，在php5x+client4x+mysql4x的新环境下，$conn=mysql\_connect(...)返回的$conn有部分情况下不可用。需要书写为：  
  
mysql\_close($conn);  
  
$conn=mysql\_connect(...);  
  
返回的$conn才可以正常使用。原因未明。未做深入研究，也未见相关讨论。或许mysql官方的BUG汇报中会有吧。  
  
～～呵呵～～  
  
  
本文来自CSDN博客，转载请标明出处：http://blog.csdn.net/brightsnow/archive/2009/03/17/3997705.aspx  
  
  
description:   
remember that your MySQL "max\_allowed\_packet" configuration setting (default 1MB)   
mysql 默认最大能够处理的是1MB   
如果你在sql使用了大的text或者BLOB数据，就会出现这个问题。 php手册上的注释   
  
When trying to INSERT or UPDATE and trying to put a large amount of text or data (blob) into a mysql table you might run into problems.   
In mysql.err you might see:   
Packet too large (73904)   
To fix you just have to start up mysql with the option -O max\_allowed\_packet=maxsize   
You would just replace maxsize with the max size you want to insert, the default is 65536   
  
  
mysql手册上说   
  
Both the client and the server have their own max\_allowed\_packet variable, so if you want to handle big packets, you must increase this variable both in the client and in the server.   
  
If you are using the mysql client program, its default max\_allowed\_packet variable is 16MB. To set a larger value, start mysql like this:   
  
shell> mysql --max\_allowed\_packet=32M That sets the packet size to 32MB.   
  
The server's default max\_allowed\_packet value is 1MB. You can increase this if the server needs to handle big queries (for example, if you are working with big BLOB columns). For example, to set the variable to 16MB, start the server like this:   
  
shell> mysqld --max\_allowed\_packet=16M You can also use an option file to set max\_allowed\_packet. For example, to set the size for the server to 16MB, add the following lines in an option file:   
  
[mysqld]max\_allowed\_packet=16M   
  
使用mysql做数据库还原的时候，由于有些数据很大，会出现这样的错误：The MySQL Server returned this Error:MySQL Error Nr.2006-MySQL server has gone away。我的一个150mb的备份还原的时候就出现了这错误。解决的方法就是找到mysql安装目录，找到my.ini文件，在文件的最后添加：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。