## 1.实战

准备工作：

1.两个虚拟机：我这里用的是CentOS5.5，IP地址分别是192.168.1.101 和192.168.1.105；

　　　　　　　101做主服务器，105做从服务器（都已经安装相同版本的Mysql）；

2.本机环境：Apache+PHP+MySQL

好了，现在开始吧，来看看这听起来高大上的主从复制是怎么回事。

原理：mysql要做到主从复制，其实依靠的是二进制日志，即：假设主服务器叫A，从服务器叫B；主从复制就是

　　   B跟着A学，A做什么，B就做什么。那么B怎么同步A的动作呢？现在A有一个日志功能，把自己所做的增删改查的动作

　　   全都记录在日志中，B只需要拿到这份日志，照着日志上面的动作施加到自己身上就可以了。这样就实现了主从复制。

扩展：MYSQL还有一种日志叫做：慢日志

　　   可以设置一个时间，那么所有执行时间超过这个时间的SQL都会被记录下来。这样就可以通过慢日志快速的找到网站中SQL的瓶颈来进行优化。

         大家有时间可以研究一下，这里不多做介绍。

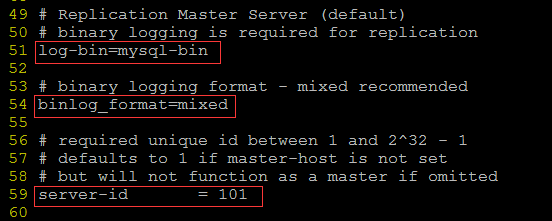
实现步骤：

1.首先修改mysql的配置文件，使其支持二进制日志功能。

打开主服务器的mysql配置文件：my.conf

代码：# vi /etc/my.cnf

加入如下三行代码：



参数解释：log-bin=mysql-bin  //将mysql二进制日志取名为mysql-bin

　　　　   binlog\_format=mixed //二进制日志的格式，有三种：statement/row/mixed,

1. statement：会将对数据库操作的sql语句写入到binlog中。
2. row：会将每一条数据的变化写入到binlog中。
3. mixed：statement与row的混合。Mysql决定什么时候写statement格式的，什么时候写row格式的binlog。

　　　　   server-id=101 //为服务器设置一个独一无二的id便于区分，这里使用ip地址的最后一位充当server-id

配置完成，:wq 保存，重启mysql

重启mysql命令：# service mysqld restart

同样的，进入从服务器，配置从服务器的my.cnf,重复步骤1即可，

唯一的区别是，server-id要改成从服务器的ip尾位，即server-id=105;其他两项是一样的，保存，并重启mySQL；

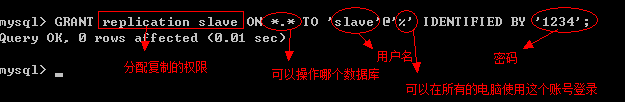
2.在主服务器上为从服务器分配一个账号，就像一把钥匙，从服务器拿着这个钥匙，才能到主服务器上来共享主服务器的日志文件。

进入主服务器的mysql界面，

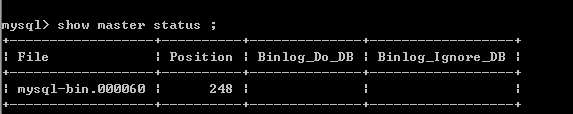
命令： # mysql -u root -p 111111     //我这里mysql账号是root，密码是111111

在mysql操作界面下，输入下面一行命令：

GRANT replication slave ON \*.\* TO 'slave'@'%' IDENTIFIED BY '111111';



1. 查看主服务器BIN日志的信息（执行完之后记录下这两值，然后在配置完从服务器之前不要对主服务器进行任何操作，因为每次操作数据库时这两值会发生改变）.



 记录mysql-bin.000060和248这两值；

4.设置从服务器

进入从服务器mysql

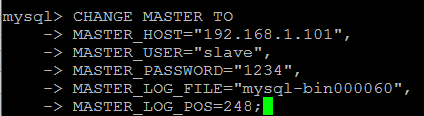
命令： # mysql -u root -p111111

关闭slave（如果你以前配置过主从的话，一定要先关闭）

命令：stop slave;

开始配置：

输入下面代码即可：



参数解释：MASTER\_HOST  :  设置要连接的主服务器的ip地址

　　　　   MASTER\_USER  :  设置要连接的主服务器的用户名

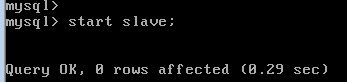
　　　　   MASTER\_PASSWORD  :  设置要连接的主服务器的密码

   　　　　MASTER\_LOG\_FILE  :  设置要连接的主服务器的bin日志的日志名称，即第3步得到的信息

   　　　　MASTER\_LOG\_POS  :  设置要连接的主服务器的bin日志的记录位置，即第3步得到的信息，（这里注意，最后一项不需要加引号。否则配置失败）

先在从服务器配置完成，启动从服务器：

命令： start slave;



5.查看是否配置成功：

命令： show slave status;

http://images2015.cnblogs.com/blog/896490/201603/896490-20160311221434663-456558902.png

上面两项均为yes，说明配置成功，否则，请重复前面的步骤。

## 2.理论

**1、什么是mysql主从同步？**

当master(主)库的数据发生变化的时候，变化会实时的同步到slave(从)库。

**2、主从同步有什么好处？**

1. 水平扩展数据库的负载能力。
2. 容错，高可用。Failover(失败切换)/High Availability
3. 数据备份。

**3、主从同步的原理是什么？**

首先我们来了解master-slave的体系结构。

如下图：



不管是delete、update、insert，还是创建函数、存储过程，所有的操作都在master上。当master有操作的时候,slave会快速的接收到这些操作，从而做同步。

**但是，这个机制是怎么实现的呢？**

在master机器上，主从同步事件会被写到特殊的log文件中(binary-log);在slave机器上，slave读取主从同步事件，并根据读取的事件变化，在slave库上做相应的更改。

如此，就实现了主从同步了！

下面我们来详细的了解。

**3.1主从同步事件有哪些**

上面说到：

在master机器上，主从同步事件会被写到特殊的log文件中(binary-log);

主从同步事件有3种形式:statement、row、mixed。

1. statement：会将对数据库操作的sql语句写入到binlog中。
2. row：会将每一条数据的变化写入到binlog中。
3. mixed：statement与row的混合。Mysql决定什么时候写statement格式的，什么时候写row格式的binlog。

3.2在master机器上的操作

当master上的数据发生改变的时候，该事件(insert、update、delete)变化会按照顺序写入到binlog中。

**binlog dump线程**

当slave连接到master的时候，master机器会为slave开启binlog dump线程。当master 的 binlog发生变化的时候，binlog dump线程会通知slave，并将相应的binlog内容发送给slave。

**3.3在slave机器上的操作**

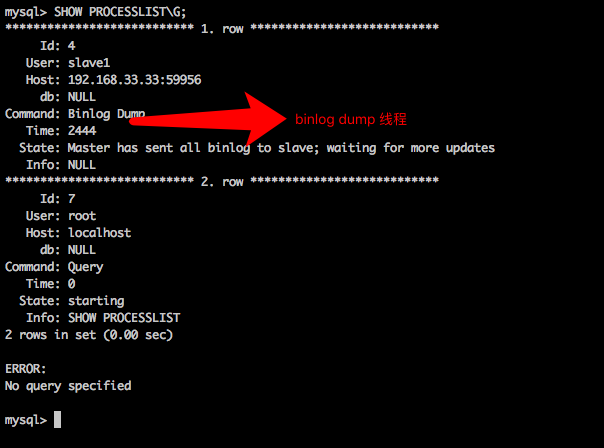
当主从同步开启的时候，slave上会创建2个线程。

1. I/O线程。该线程连接到master机器，master机器上的binlog dump线程会将binlog的内容发送给该I/O线程。该I/O线程接收到binlog内容后，再将内容写入到本地的relay log。
2. SQL线程。该线程读取I/O线程写入的relay log。并且根据relay log的内容对slave数据库做相应的操作。

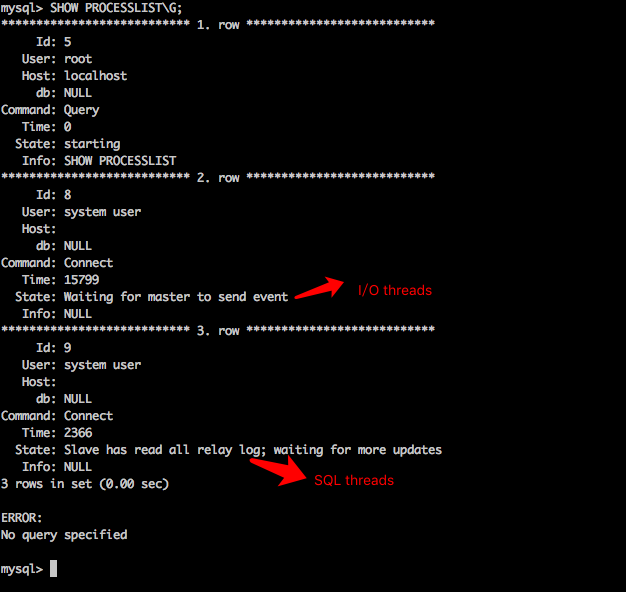
3.4如何在master、slave上查看上述的线程？

使用SHOW PROCESSLIST命令可以查看。

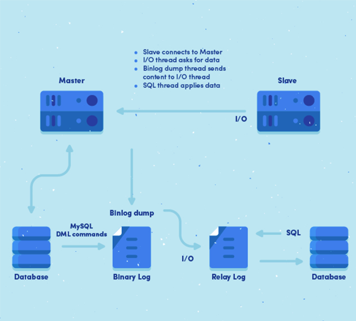
如图，在master机器上查看binlog dump线程。



如图，在slave机器上查看I/O、SQL线程。



**4、讲了这么多，一图以蔽之**



## ecshop改造读写主从数据库分离 实例

http://www.uuecs.com/article-shouce/467.html

|  |
| --- |
| data/config.php    <?php    $db\_name   = "<a href="http://www.uuecs.com/" target="\_blank" title="ecshop"><strong class="keylink">ecshop</strong></a>";    $prefix    = "ecs\_";    $timezone    = "Europe/Berlin";    $cookie\_path    = "/";    $cookie\_domain    = "";    $session = "1440";    $\_config = array();    //数据库主服务器设置, 支持多组服务器设置, 当设置多组服务器时, 则会随机使用某个服务器  $\_config['master'][1]['dbhost'] = "192.168.2.175:3306";  $\_config['master'][1]['dbname'] = "ecshop";  $\_config['master'][1]['dbuser'] = "dragon";  $\_config['master'][1]['dbpw'] = "loong";    /\*   \*$\_config['master'][2]['dbhost'] = "";   \*...   \*/    //数据库从服务器设置( slave, 只读 ), 支持多组服务器设置, 当设置多组服务器时, 系统每次随机使用  $\_config['slave'][1]['dbhost'] = "192.168.2.176:3306";  $\_config['slave'][1]['dbname'] = "ecshop";  $\_config['slave'][1]['dbuser'] = "ivan";  $\_config['slave'][1]['dbpw'] = "loong";    $\_config['slave'][2]['dbhost'] = "192.168.2.177:3306";  $\_config['slave'][2]['dbname'] = "ecshop";  $\_config['slave'][2]['dbuser'] = "ivan";  $\_config['slave'][2]['dbpw'] = "loong";    define('EC\_CHARSET','utf-8');    define('ADMIN\_PATH','admin');    define('AUTH\_KEY', 'this is a key');    define('OLD\_AUTH\_KEY', '');    define('API\_TIME', '');    ?>    初始化数据连接类          /\* 初始化数据库类       \* 如果配置了从服务器，则初始化从库类      \*/      if(count($\_config['slave'])) {          require(ROOT\_PATH . 'includes/cls\_mysql\_slave.php');          $db = new cls\_mysql\_slave($\_config);      }else{          require(ROOT\_PATH . 'includes/cls\_mysql.php');          $db = new cls\_mysql($\_config);      }  增加cls\_mysql\_slave.php从库类      <?php    require(ROOT\_PATH . 'includes/cls\_mysql.php');  class cls\_mysql\_slave extends cls\_mysql  {   var $slaveid = null;     function set\_config($config){   if(!empty($this->config['slave'])) {   $this->slaveid = array\_rand($this->config['slave']);   }   parent::set\_config($config);   }     /\* 随机分配从库连接 \*/   function set\_slave\_config() {   $this->settings = $this->config['slave'][$this->slaveid];   $this->settings['charset'] = $this->config['charset'];   $this->settings['pconnect'] = $this->config['pconnect'];   }     function slave\_connect() {   $this->set\_slave\_config();   $dbhost = $this->settings['dbhost'];   $dbuser = $this->settings['dbuser'];   $dbpw = $this->settings['dbpw'];   $dbname = $this->settings['dbname'];   $this->connect($dbhost, $dbuser, $dbpw, $dbname);     }       function query($sql, $type = '') {   // 如果执行查询操作，则执行从库连接   if($this->slaveid && strtoupper(substr($sql, 0 , 6)) == 'SELECT') {   $this->slave\_connect();   }else{   parent::set\_config($this->config);   $dbhost = $this->settings['dbhost'];   $dbuser = $this->settings['dbuser'];   $dbpw = $this->settings['dbpw'];   $dbname = $this->settings['dbname'];   $this->connect($dbhost, $dbuser, $dbpw, $dbname);   }   return parent::query($sql, $type);   }     /\* 删除失败连接\*/   function del\_error\_link(){   unset($this->config['slave'][$this->slaveid]);   $this->set\_config($this->config);   $this->set\_slave\_config();   $dbhost = $this->settings['dbhost'];   $dbuser = $this->settings['dbuser'];   $dbpw = $this->settings['dbpw'];   $dbname = $this->settings['dbname'];   $this->connect($dbhost, $dbuser, $dbpw, $dbname);   }    }  cls\_mysql.php文件类修改      <?php    if (!defined('IN\_ECS'))  {      die('Hacking attempt');  }    class cls\_mysql  {      var $link\_id    = NULL;        var $settings   = array();        var $queryCount = 0;      var $linkCount = 0;      var $queryTime  = '';      var $queryLog   = array();        var $max\_cache\_time = 300; // 最大的缓存时间，以秒为单位        var $cache\_data\_dir = 'temp/query\_caches/';      var $root\_path      = '';        var $error\_message  = array();      var $platform       = '';      var $version        = '';      var $dbhash         = '';      var $starttime      = 0;      var $timeline       = 0;      var $timezone       = 0;        var $mysql\_config\_cache\_file\_time = 0;        var $mysql\_disable\_cache\_tables = array(); // 不允许被缓存的表，遇到将不会进行缓存      var $config = array();        function \_\_construct($config, $charset = 'utf8', $pconnect = 0, $quiet = 0)      {          $this->cls\_mysql($config, $charset, $pconnect, $quiet);      }        function cls\_mysql($config, $charset = 'utf8', $pconnect = 0, $quiet = 0)      {          if(!empty($config)) {              $config['charset'] = $charset;              $config['pconnect'] = $pconnect;              $this->config = $config;          }            if (defined('EC\_CHARSET'))          {              $charset = strtolower(str\_replace('-', '', EC\_CHARSET));          }            if (defined('ROOT\_PATH') && !$this->root\_path)          {              $this->root\_path = ROOT\_PATH;          }            $this->set\_config($this->config);            if ($quiet)          {              $dbhost = $this->settings['dbhost'];              $dbuser = $this->settings['dbuser'];              $dbpw = $this->settings['dbpw'];              $dbname = $this->settings['dbname'];              $this->connect($dbhost, $dbuser, $dbpw, $dbname, $charset, $pconnect, $quiet);          }      }        //随机分配数据库连接      function set\_config($config) {          $sid = array\_rand($config['master']);          $settings = $config['master'][$sid];          $settings['sid'] = $sid;//www.zuimoban.com          $settings['charset'] = $this->config['charset'];          $settings['pconnect'] = $this->config['pconnect'];          $this->settings = $settings;      }        function connect($dbhost, $dbuser, $dbpw, $dbname = '', $charset = 'utf8', $pconnect = 0, $quiet = 0)      {          if ($pconnect)          {              if (!($this->link\_id = @mysql\_pconnect($dbhost, $dbuser, $dbpw)))              {                  if (!$quiet)                  {                      $this->ErrorMsg("Can't pConnect MySQL Server!");                  }                    return false;              }          }          else          {              if (PHP\_VERSION >= '4.2')              {                  $this->link\_id = @mysql\_connect($dbhost, $dbuser, $dbpw, true);              }              else              {                  $this->link\_id = @mysql\_connect($dbhost, $dbuser, $dbpw);                    mt\_srand((double)microtime() \* 1000000); // 对 PHP 4.2 以下的版本进行随机数函数的初始化工作              }              if (!$this->link\_id)              {                  if (!$quiet)                  {                      //连接超过10次，中断连接，抛出错误                      if($this->linkCount>9){                          $this->ErrorMsg("Can't Connect MySQL Server!");                      }                      $this->linkCount++;                      $this->del\_error\_link();                  }                    return false;              }          }            $this->dbhash  = md5($this->root\_path . $dbhost . $dbuser . $dbpw . $dbname);          $this->version = mysql\_get\_server\_info($this->link\_id);            /\* 如果mysql 版本是 4.1+ 以上，需要对字符集进行初始化 \*/          if ($this->version > '4.1')          {              if ($charset != 'latin1')              {                  mysql\_query("SET character\_set\_connection=$charset, character\_set\_results=$charset, character\_set\_client=binary", $this->link\_id);              }              if ($this->version > '5.0.1')              {                  mysql\_query("SET sql\_mode=''", $this->link\_id);              }          }            $sqlcache\_config\_file = $this->root\_path . $this->cache\_data\_dir . 'sqlcache\_config\_file\_' . $this->dbhash . '.php';            @include($sqlcache\_config\_file);            $this->starttime = time();            if ($this->max\_cache\_time && $this->starttime > $this->mysql\_config\_cache\_file\_time + $this->max\_cache\_time)          {              if ($dbhost != '.')              {                  $result = mysql\_query("SHOW VARIABLES LIKE 'basedir'", $this->link\_id);                  $row    = mysql\_fetch\_assoc($result);                  if (!empty($row['Value']{1}) && $row['Value']{1} == ':' && !empty($row['Value']{2}) && $row['Value']{2} == "\\")                  {                      $this->platform = 'WINDOWS';                  }                  else                  {                      $this->platform = 'OTHER';                  }              }              else              {                  $this->platform = 'WINDOWS';              }                if ($this->platform == 'OTHER' &&                  ($dbhost != '.' && strtolower($dbhost) != 'localhost:3306' && $dbhost != '127.0.0.1:3306') ||                  (PHP\_VERSION >= '5.1' && date\_default\_timezone\_get() == 'UTC'))              {                  $result = mysql\_query("SELECT UNIX\_TIMESTAMP() AS timeline, UNIX\_TIMESTAMP('" . date('Y-m-d H:i:s', $this->starttime) . "') AS timezone", $this->link\_id);                  $row    = mysql\_fetch\_assoc($result);                    if ($dbhost != '.' && strtolower($dbhost) != 'localhost:3306' && $dbhost != '127.0.0.1:3306')                  {                      $this->timeline = $this->starttime - $row['timeline'];                  }                    if (PHP\_VERSION >= '5.1' && date\_default\_timezone\_get() == 'UTC')                  {                      $this->timezone = $this->starttime - $row['timezone'];                  }              }                $content = '<' . "?php\r\n" .                         '$this->mysql\_config\_cache\_file\_time = ' . $this->starttime . ";\r\n" .                         '$this->timeline = ' . $this->timeline . ";\r\n" .                         '$this->timezone = ' . $this->timezone . ";\r\n" .                         '$this->platform = ' . "'" . $this->platform . "';\r\n?" . '>';                @file\_put\_contents($sqlcache\_config\_file, $content);          }            /\* 选择数据库 \*/          if ($dbname)          {              if (mysql\_select\_db($dbname, $this->link\_id) === false )              {                  if (!$quiet)                  {                      $this->ErrorMsg("Can't select MySQL database!");                  }                    return false;              }              else              {                  return true;              }          }          else          {              return true;          }      }        ......        /\* 删除失败连接\*/      function del\_error\_link(){          unset($this->config['master'][$this->settings['sid']]);          $this->set\_config($this->config);          $dbhost = $this->settings['dbhost'];          $dbuser = $this->settings['dbuser'];          $dbpw = $this->settings['dbpw'];          $dbname = $this->settings['dbname'];          $this->connect($dbhost, $dbuser, $dbpw, $dbname);      }  } |

## 读写分离实例（可用）

|  |
| --- |
| <?php /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\* mysql-rw-php version 0.1 @ 2009-4-16 \*\*\* code by hqlulu#gmail.com \*\*\* http://www.aslibra.com \*\*\* http://code.google.com/p/mysql-rw-php/ \*\*\* code modify from class\_mysql.php (uchome) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ class mysql\_rw\_php {   //查询个数   var $querynum = 0;   //当前操作的数据库连接   var $link = null;   //字符集   var $charset;   //当前数据库   var $cur\_db = '';   //是否存在有效的只读数据库连接   var $ro\_exist = false;   //只读数据库连接   var $link\_ro = null;   //读写数据库连接   var $link\_rw = null;   function mysql\_rw\_php(){   }   function connect($dbhost, $dbuser, $dbpw, $dbname = '', $pconnect = 0, $halt = TRUE) {     if($pconnect) {       if(!$this->link = @mysql\_pconnect($dbhost, $dbuser, $dbpw)) {         $halt && $this->halt('Can not connect to MySQL server');       }     } else {       if(!$this->link = @mysql\_connect($dbhost, $dbuser, $dbpw)) {         $halt && $this->halt('Can not connect to MySQL server');       }     }          //只读连接失败     if(!$this->link && !$halt) return false;          //未初始化rw时，第一个连接作为rw     if($this->link\_rw == null)       $this->link\_rw = $this->link;     if($this->version() > '4.1') {       if($this->charset) {         @mysql\_query("SET character\_set\_connection=$this->charset, character\_set\_results=$this->charset, character\_set\_client=binary", $this->link);       }       if($this->version() > '5.0.1') {         @mysql\_query("SET sql\_mode=''", $this->link);       }     }     if($dbname) {       $this->select\_db($dbname);     }   }   //连接一个只读的mysql数据库   function connect\_ro($dbhost, $dbuser, $dbpw, $dbname = '', $pconnect = 0){     if($this->link\_rw == null)       $this->link\_rw = $this->link;     $this->link = null;     //不产生halt错误     $this->connect($dbhost, $dbuser, $dbpw, $dbname, $pconnect, false);     if($this->link){       //连接成功       //echo "link ro sussess!<br>";       $this->ro\_exist = true;       $this->link\_ro = $this->link;       if($this->cur\_db){         //如果已经选择过数据库则需要操作一次         @mysql\_select\_db($this->cur\_db, $this->link\_ro);       }     }else{       //连接失败       //echo "link ro failed!<br>";       $this->link = &$this->link\_rw;     }   }   //设置一系列只读数据库并且连接其中一个   function set\_ro\_list($ro\_list){     if(is\_array($ro\_list)){       //随机选择其中一个       $link\_ro = $ro\_list[array\_rand($ro\_list)];       $this->connect\_ro($link\_ro['dbhost'], $link\_ro['dbuser'], $link\_ro['dbpw']);     }   }   function select\_db($dbname) {     //同时操作两个数据库连接     $this->cur\_db = $dbname;     if($this->ro\_exist){       @mysql\_select\_db($dbname, $this->link\_ro);     }     return @mysql\_select\_db($dbname, $this->link\_rw);   }   function fetch\_array($query, $result\_type = MYSQL\_ASSOC) {     return mysql\_fetch\_array($query, $result\_type);   }   function fetch\_one\_array($sql, $type = '') {     $qr = $this->query($sql, $type);     return $this->fetch\_array($qr);   }   function query($sql, $type = '') {     $this->link = &$this->link\_rw;     //判断是否select语句     if($this->ro\_exist && preg\_match ("/^(\s\*)select/i", $sql)){       $this->link = &$this->link\_ro;     }     $func = $type == 'UNBUFFERED' && @function\_exists(\'mysql\_unbuffered\_query\') ?       'mysql\_unbuffered\_query' : 'mysql\_query';     if(!($query = $func($sql, $this->link)) && $type != 'SILENT') {       $this->halt('MySQL Query Error', $sql);     }     $this->querynum++;     return $query;   }   function affected\_rows() {     return mysql\_affected\_rows($this->link);   }   function error() {     return (($this->link) ? mysql\_error($this->link) : mysql\_error());   }   function errno() {     return intval(($this->link) ? mysql\_errno($this->link) : mysql\_errno());   }   function result($query, $row) {     $query = @mysql\_result($query, $row);     return $query;   }   function num\_rows($query) {     $query = mysql\_num\_rows($query);     return $query;   }   function num\_fields($query) {     return mysql\_num\_fields($query);   }   function free\_result($query) {     return mysql\_free\_result($query);   }   function insert\_id() {     return ($id = mysql\_insert\_id($this->link)) >= 0 ? $id : $this->result($this->query("SELECT last\_insert\_id()"), 0);   }   function fetch\_row($query) {     $query = mysql\_fetch\_row($query);     return $query;   }   function fetch\_fields($query) {     return mysql\_fetch\_field($query);   }   function version() {     return mysql\_get\_server\_info($this->link);   }   function close() {     return mysql\_close($this->link);   }   function halt($message = '', $sql = '') {     $dberror = $this->error();     $dberrno = $this->errno();     echo "<div style=\"position:absolute;font-size:11px;font-family:verdana,arial;background:#EBEBEB;padding:0.5em;\">         <b>MySQL Error</b><br>         <b>Message</b>: $message<br>         <b>SQL</b>: $sql<br>         <b>Error</b>: $dberror<br>         <b>Errno.</b>: $dberrno<br>         </div>";     exit();   } } ?> |

example.php

|  |
| --- |
| <?php /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\* mysql-rw-php version 0.1 @ 2009-4-16 \*\*\* code by hqlulu#gmail.com \*\*\* http://www.aslibra.com \*\*\* http://code.google.com/p/mysql-rw-php/ \*\*\* code modify from class\_mysql.php (uchome) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ require\_once('mysql\_rw\_php.class.php'); //rw info $db\_rw = array(   'dbhost'=>'www.aslibra.com',   'dbuser'=>'aslibra',   'dbpw'=>'www.aslibra.com',   'dbname'=>'test' ); $db\_ro = array(   array(     'dbhost'=>'www.aslibra.com:4306',     'dbuser'=>'aslibra',     'dbpw'=>'www.aslibra.com'   ) ); $DB = new mysql\_rw\_php; //connect Master $DB->connect($db\_rw[dbhost], $db\_rw[dbuser], $db\_rw[dbpw], $db\_rw[dbname]); //Method 1: connect one server $DB->connect\_ro($db\_ro[0][dbhost], $db\_ro[0][dbuser], $db\_ro[0][dbpw]); //Method 2: connect one server from a list by rand $DB->set\_ro\_list($db\_ro); //send to rw $sql = "insert into a set a='test'"; $DB->query($sql); //send to ro $sql = "select \* from a"; $qr = $DB->query($sql); while($row = $DB->fetch\_array($qr)){   echo $row[a]; } ?> |