目录

[一、判断大型网站的标准： 1](#_Toc402279541)

[1、pv(page views)网页的浏览量： 1](#_Toc402279542)

[2、uv值（unique vistor）独立访客 1](#_Toc402279543)

[3、独立ip， 1](#_Toc402279544)

[二、大型网站带来的一些问题。 1](#_Toc402279545)

[1、大的并发。 1](#_Toc402279546)

[2、大流量。 2](#_Toc402279547)

[3、大的存储， 2](#_Toc402279548)

[三、大并发的解决方案： 2](#_Toc402279549)

[1、负载均衡器： 2](#_Toc402279550)

[2、负载均衡实现的方式： 3](#_Toc402279551)

[3、集群： 3](#_Toc402279552)

[四、大流量解决方案： 3](#_Toc402279553)

[1、防止我们的网站资源被盗链。 3](#_Toc402279554)

[2、减少http请求， 3](#_Toc402279555)

[3、启用压缩， 6](#_Toc402279556)

[4、通过浏览器缓存数据内容。 7](#_Toc402279557)

[5、可以把比较占用流量的一些资源，单独组建一个服务器， 7](#_Toc402279558)

[6、花钱买带宽。 8](#_Toc402279559)

[五、大存储解决方案： 8](#_Toc402279560)

[1、缓存技术： 8](#_Toc402279561)

[2、在设计表的时候，要满足3范式 9](#_Toc402279562)

[3、要给表添加适当的索引：索引非常重要的，可以提高查询速度。 9](#_Toc402279563)

[4、要创建适当的存储过程，函数，触发器等。 10](#_Toc402279564)

[5、读写分离（主从服务器） 10](#_Toc402279565)

[6、分表技术（垂直分割和水平分割） 10](#_Toc402279566)

[7、分区技术 10](#_Toc402279567)

[8、升级mysql服务器（添加配置：加大内容，64位） 11](#_Toc402279568)

[9、要对sql语句进行调优 11](#_Toc402279569)

[10、对配置文件进行优化配置。 11](#_Toc402279570)

[六、网页静态化的内容： 11](#_Toc402279571)

[1、网页静态化的使用原理： 12](#_Toc402279572)

[2、了解几个概念： 12](#_Toc402279573)

[3、实现静态化的方式： 13](#_Toc402279574)

[4、通过ob缓存来实现真静态 13](#_Toc402279575)

[5、利用ob缓存完成一个真静态案例 16](#_Toc402279576)

[6、案例扩展，要给生成对应的静态页面一个有效期。 17](#_Toc402279577)

[七、apache并发工具测试。 19](#_Toc402279578)

[1、ab.exe测试 19](#_Toc402279579)

[2、要调整apache的并发配置，要清楚处理多并发的方式。 20](#_Toc402279580)

[3、如何知道，当前apache服务器是使用哪一种方式处理并发 22](#_Toc402279581)

[4、如何配置MPM(多路处理模块)的问题，调整最大的并发量。 22](#_Toc402279582)

# 一、判断大型网站的标准：

## 1、pv(page views)网页的浏览量：

概念：一个网站所有的页面，在24小时内被访问的总的次数。千万级别，百万级别，

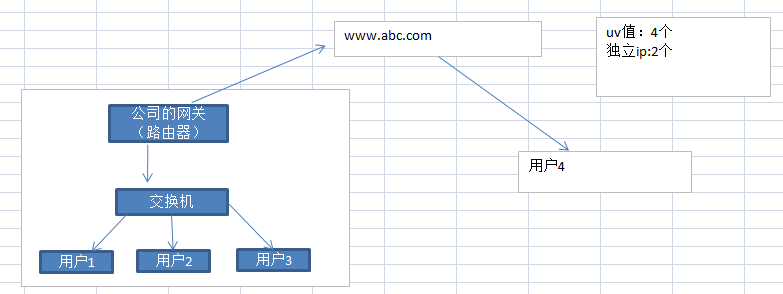
## 2、uv值（unique vistor）独立访客

概念：一个网站，在24小时内，有多少个用户来访问我们的网站。达到10万

## 3、独立ip，

概念：一个网站，在24小时内，有多少个ip来访问我们的网站。

uv值约等于独立ip.如果要考虑局域网，uv值略大于独立ip



# 二、大型网站带来的一些问题。

## 1、大的并发。

并发量：在同一时间点（1秒内），有多少个用户同时访问我们的网站。对同一个网址，同时刷新浏览器。达到500，就非常大了。

假如并发量是500，pv值是多少。500\*3600\*10

## 2、大流量。

网站需要的大的带宽。10G.

## 3、大的存储，

网站中的数据库，表的容量成海量趋势，GT级别，如何快速的查找出想要的数据。

# 三、大并发的解决方案：

要对网站的服务器重新架构，采用分层，负载均衡的架构。

## 1、负载均衡器：

硬件：f5-bigip 性能比较好，立竿见影，价格昂贵，一般适合于大型网站公司，网游公司。



软件：

lvs(linux virtual server)linux虚拟服务，加入到linux的内核中。

nginx:可以做web服务器（apache）,还可以做负载均衡。

## 2、负载均衡实现的方式：

主要有：

轮询技术：客户端请求服务器轮流转发。

ip哈希：同一ip地址的客户端，始终请求同一台服务器。

最少连接：把请求转发给最空闲的服务器。

## 3、集群：

主要是解决计算机单点故障，在一个集群中的计算机，只有一台计算机工作，其他计算机处于休眠状态，监视正在工作的计算机，当正在工作的计算机出现问题，则休眠的计算机立刻接替工作。

# 四、大流量解决方案：

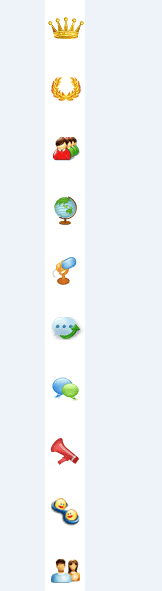
## 1、防止我们的网站资源被盗链。

可以采用一些非技术手段防止被盗链，在图片上添加水印

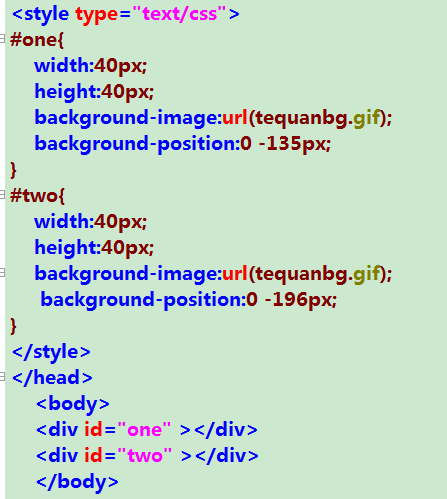


## 2、减少http请求，

主要手段就是合并js文件,css文件，背景图片的文件。将浏览器需要的样式文件或者js文件，合并成一个样式文件或js文件。比如通过背景图片举个例子。



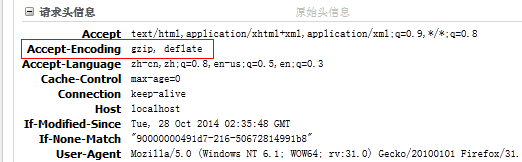
具体的代码：





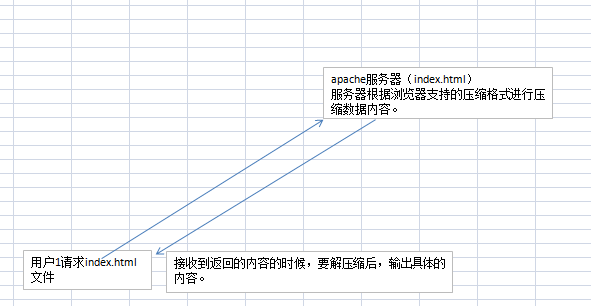
## 3、启用压缩，

减少数据传输的数据量，常见的压缩格式是：gzip,deflate.

，

告诉服务器当前浏览器支持的压缩的格式

原理：

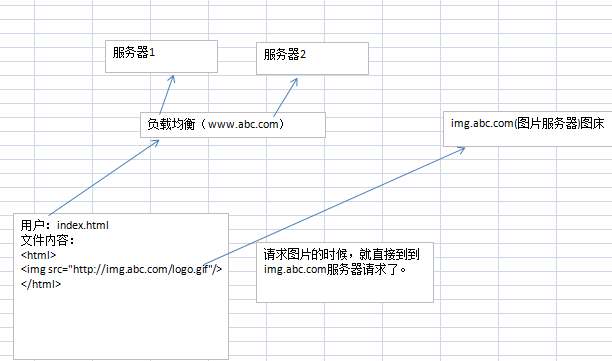


## 4、通过浏览器缓存数据内容。

在网站中有一些资源，比如js文件，css文件，一些图片文件，更新的频率比较少。通过个设置http的cache-control expires属性来进行设置缓存，可以设置缓存的文件类型，设置缓存的缓存周期，

## 5、可以把比较占用流量的一些资源，单独组建一个服务器，

比如图片服务器，视频服务器等。



要注意：资源服务器的配置：

存储资源的服务器：主要要求是硬盘的容量，读写速度。

可以组建磁盘阵列。

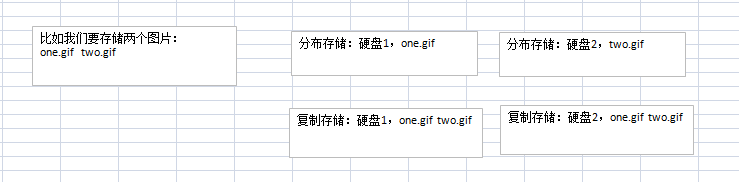
raid0

raid1

磁盘阵列的存储技术：

分布存储：至少是两块硬盘

复制存储：至少是两块硬盘



## 6、花钱买带宽。



# 五、大存储解决方案：

## 1、缓存技术：

通过缓存技术，达到不查询数据库或者少查询数据库的目的。

计算机的访问速度，内存》硬盘文件》数据库

缓存技术主要有：

磁盘缓存（页面静态化），把一个查询数据库的页面变成一个不查询数据库的页面

内存缓存：把经常查询的数据保存到内存里面，下次查询数据时候直接在 内存里面查询。

（memcache/redis/mysql的memory引擎）

## 2、在设计表的时候，要满足3范式

第一范式是：原子性，字段不能再分割了。只要是关系型数据库就自动满足第一范式：

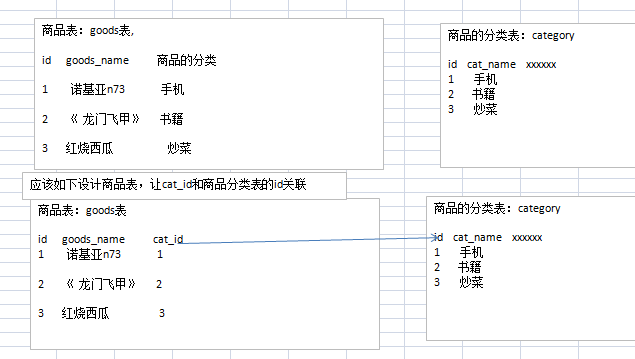
数据库的分类：

关系型数据库：有行和 列的概念，二维表格。常见的关系型数据库：mysql,sql server,oracle,db2,

非关系型数据库（nosql）面向集合和 文档的，没有行和列的概念常见的有redis/mongodb等。

第二范式：在一个表中不能有完全相同的记录。可以通过设置一个主键。

第三范式：表中的字段不能冗余存储。



## 3、要给表添加适当的索引：索引非常重要的，可以提高查询速度。

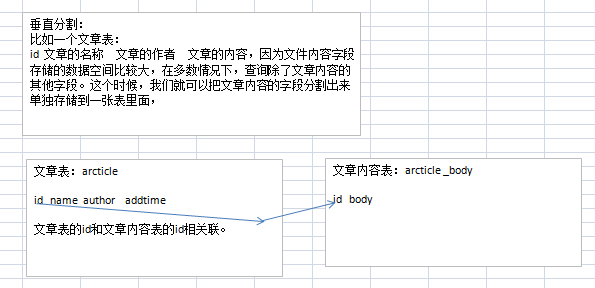
常见索引有：主键索引，唯一索引，普通索引，全文索引，

## 4、要创建适当的存储过程，函数，触发器等。

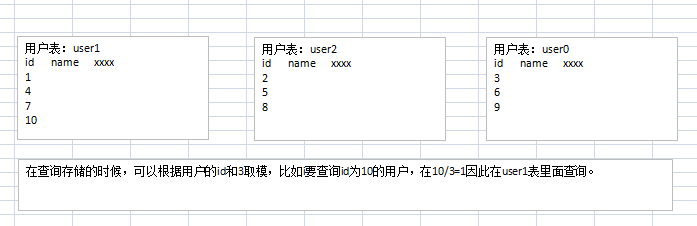
## 5、读写分离（主从服务器）

## 6、分表技术（垂直分割和水平分割）

垂直分割：



水平分割：



## 7、分区技术

把一个表的数据内容，在不同的 区域存储，

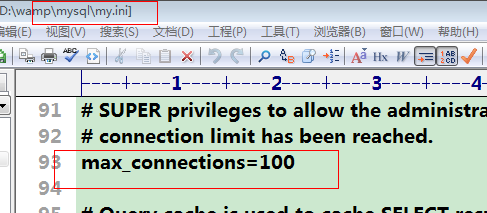
## 8、升级mysql服务器（添加配置：加大内容，64位）

## 9、要对sql语句进行调优

select \* from tablename 该语句不要使用，要按需查询。需要哪个字段的数据，就查询哪个字段的数据。

## 10、对配置文件进行优化配置。

比如配置mysql数据库的并发量：



# 六、网页静态化的内容：

主要有两种：

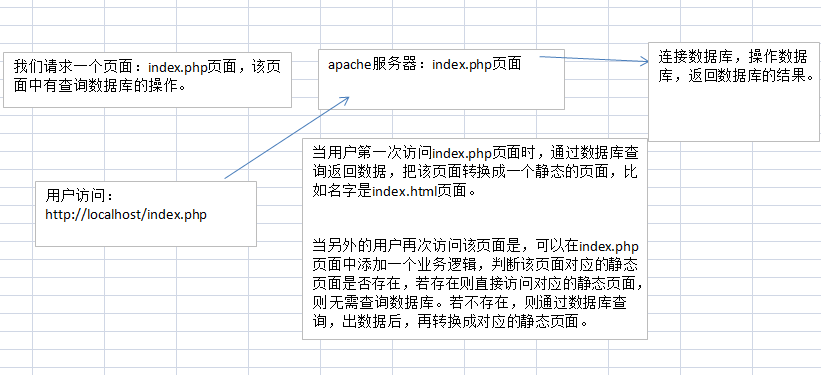
真静态:就是把一个动态（查询数据库）的页面，转换成一个静态的页面html页面，

伪静态：从形式上看url地址是一个静态的页面，实际上还是要查询数据库的。

比如：<http://www.abc.com/news-soprt-id100.html>,实际上对应

<http://www.abc.com/news.php?type=sport&id=100>页面。

## 1、网页静态化的使用原理：



## 2、了解几个概念：

动态的网址：一般来说,该页面有查询数据库的功能,比如后缀是：（php|asp|aspx|jsp）

特点：查询数据库，访问速度慢，可能sql注入，不是很安全，不是利于seo

静态的网址：一般来说就是html页面。

特点：不查询数据库，访问速度快，安全，利于seo

伪静态网址：从形式上看是一个静态网址，实际上是一个动态页面，

特点：查询数据库，安全，利于seo

比如：http://localhost/1028/demo.php/goods\_id100.html



## 3、实现静态化的方式：

真静态实现方式：

（1）通过ob缓存技术来实现。

（2）通过模板替换技术来实现

伪静态实现方式：

（1）通过正则表达式，匹配替换

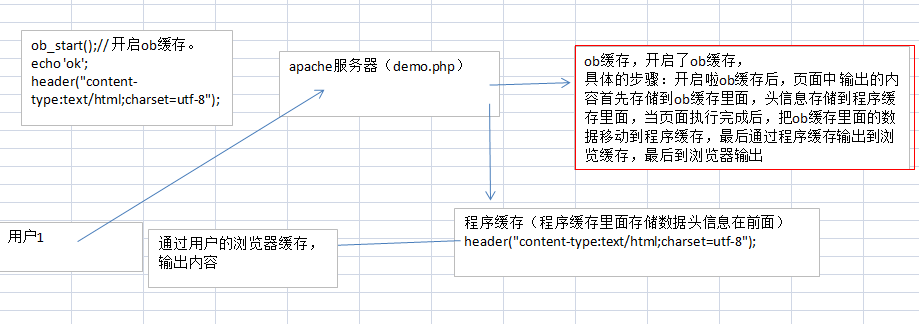
（2）通过服务器的rewrite机制。

## 4、通过ob缓存来实现真静态

（1）概念：什么是ob缓存：

ob缓存：output\_buffering:输出缓存，我们请求一个php页面，实际上是通过三个缓存的。

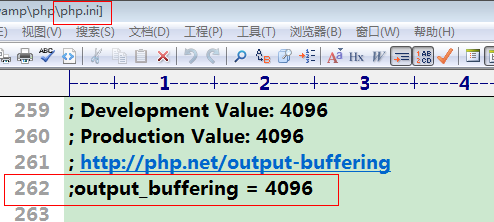
ob缓存（如果开启）---》程序缓存------》浏览器缓存，要注意：程序缓存和浏览器缓存必须存在的。



（2）如何开启ob缓存：

第一种方式：通过ob\_start()函数，只对当前页面有效

第二种方式：通过php.ini文件中：



（3）学习几个函数来讲解

ob\_clean();清除ob缓存里面的数据，并不关闭ob缓存。



ob\_end\_clean():清除ob缓存里面的数据，并关闭ob缓存



ob\_flush();把ob缓存里面的数据刷新到程序缓存，并不关闭ob缓存。



ob\_end\_flush():把ob缓存里面 的数据移动到程序缓存，并关闭ob缓存



ob\_get\_contents();获取ob缓存里面数据



## 5、利用ob缓存完成一个真静态案例

思路：判断该页面对应的静态页面，是否存在，若存在的直接访问静态页面，不存在则通过数据库查询出数据，并生成一个静态页面。

具体的代码：

<?php

header("content-type:text/html;charset=utf-8");

$filename="index.html";

//思路：判断该页面对应的静态页面，是否存在，

if(file\_exists($filename)){

include $filename;exit;

}

//连接数据库的操作：

$conn = mysql\_connect("localhost",'root','root');

mysql\_query("use itdede");

mysql\_query("set names utf8");

$sql="select title from dede\_archives limit 10";

$res = mysql\_query($sql,$conn);

$list=array();

while($row=mysql\_fetch\_assoc($res)){

$list[]=$row;

}

ob\_start();

//echo 'ok';

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="zh-CN">

<head>

<title>新建网页</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="description" content="" />

<meta name="keywords" content="" />

<script type="text/javascript">

</script>

<style type="text/css">

</style>

</head>

<body>

<table width="500" border="1" >

<tr><td>电影名称</td><td>操作</td></tr>

<?php foreach($list as $v){?>

<tr><td><?php echo $v['title']?></td><td>电影 详情</td></tr>

<?php }?>

</table>

</body>

</html>

<?PHP

$contents = ob\_get\_contents();

file\_put\_contents($filename,$contents);

?>

## 6、案例扩展，要给生成对应的静态页面一个有效期。

判断条件：对应的静态页面要存在，而且在有效期内，就可以直接读取静态页面。

如何计算在有效期内：

**对应静态页面的修改的时间戳+有效期>当前的时间戳**

filemtime($filename)+300>time();

假如一个网站并发是1000，不缓存的话。60秒之内查询数据库多少次。60000次

假如一个网站并发是1000，缓存60秒的话，60秒之内查询数据库多少次。1次

具体的代码：

<?php

header("content-type:text/html;charset=utf-8");

$filename="index.html";

//思路：判断该页面对应的静态页面，是否存在，

if(file\_exists($filename) && filemtime($filename)+10>time()){

include $filename;exit;

}

//连接数据库的操作：

$conn = mysql\_connect("localhost",'root','root');

mysql\_query("use itdede");

mysql\_query("set names utf8");

$sql="select title from dede\_archives limit 10";

$res = mysql\_query($sql,$conn);

$list=array();

while($row=mysql\_fetch\_assoc($res)){

$list[]=$row;

}

ob\_start();

echo time();

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="zh-CN">

<head>

<title>新建网页</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="description" content="" />

<meta name="keywords" content="" />

<script type="text/javascript">

</script>

<style type="text/css">

</style>

</head>

<body>

<table width="500" border="1" >

<tr><td>电影名称</td><td>操作</td></tr>

<?php foreach($list as $v){?>

<tr><td><?php echo $v['title']?></td><td>电影 详情</td></tr>

<?php }?>

</table>

</body>

</html>

<?PHP

$contents = ob\_get\_contents();

file\_put\_contents($filename,$contents);

?>

该案例重点是：理清如下思路即可。

//思路：判断该页面对应的静态页面，是否存在，

if(file\_exists($filename) && filemtime($filename)+10>time()){

include $filename;exit;

}

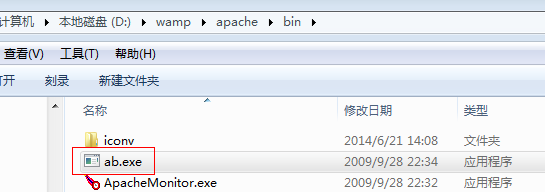
# 七、apache并发工具测试。

## 1、ab.exe测试

我们做好一个网站后，要测一下当前网站架构，能够支持并发情况。

测试工具：

可以使用apache自带的一个工具。



并发测试的工具有很多，

常用的压力测试软件：

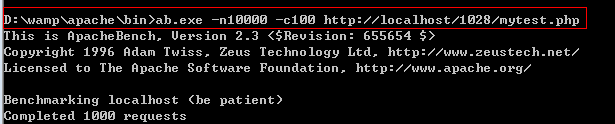
ab ： 优点：可以模拟各种请求 缺点：最大只能支撑1000的并发

webbench： 优点：30000万的 并发 缺点：只能模拟GET请求

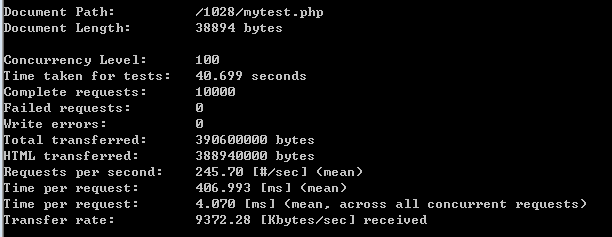
loadrunner : 非常专业的压力测试软件

winrunner:专业的压力测试软件。

ab.exe -n访问的总的次数 -c用户并发数量（有多少人同时访问）网站的地址。



测试结果：



一人请求一次需要的时间。

100个人同时请求一次，需要的时间

一秒中请求的次数

总的请求次数

总共用的时间

请求的时间越短越好。

把同时请求的人数调整到600则出现了问题。实际上，在默认情况下，apahce最高支持150并发。

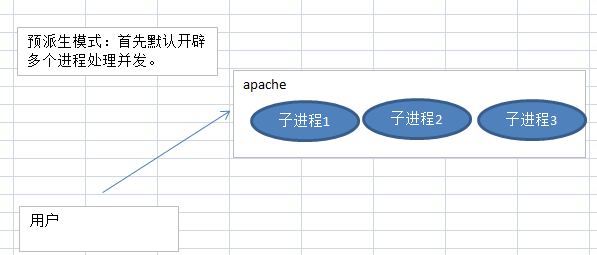


## 2、要调整apache的并发配置，要清楚处理多并发的方式。

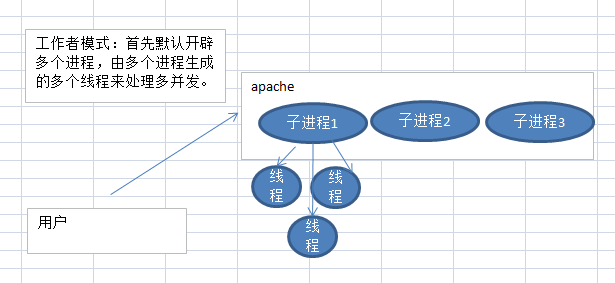
首先搞清楚当前apache是什么MPM(多路处理模块), 通俗讲就是apache处理多并发的方式,

常见的有三种

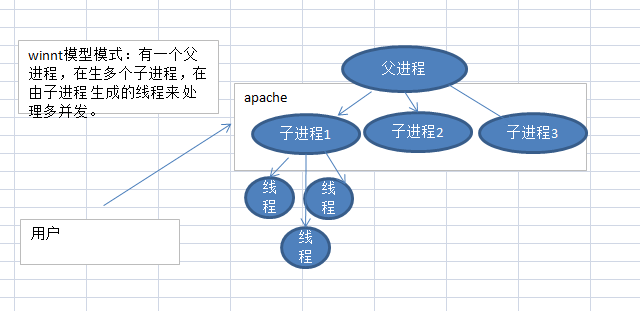
（1）perfork(预派生模式)



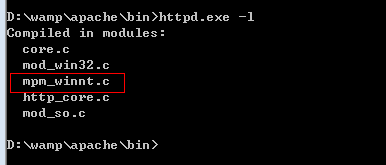
（2）worker(工作者模式)



（3）winnt模式(windows下默认的模式)



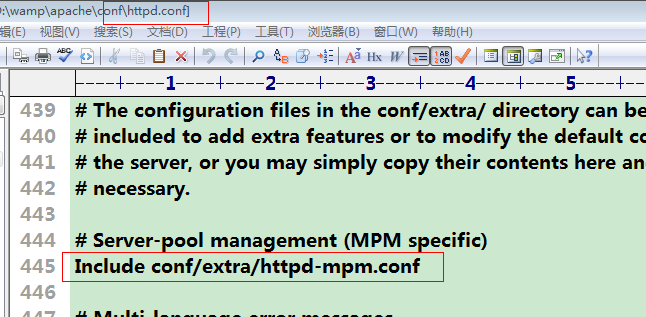
## 3、如何知道，当前apache服务器是使用哪一种方式处理并发



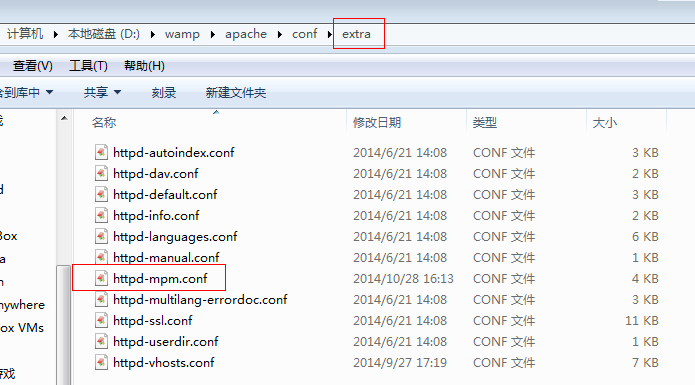
显示出当前apache处理并发的方式。

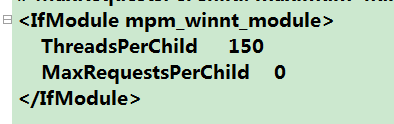
## 4、如何配置MPM(多路处理模块)的问题，调整最大的并发量。

（1）打开apache的配置文件 httpd.conf,开启多路处理模型



（2）打开extra目录下面的httpd-mpm.conf辅助配置文件。

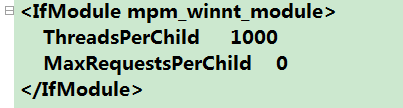




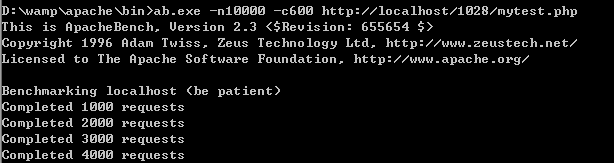
模块的指令

模块名称

（3）根据当前apache的mpm的方式进行调整如下：



测试如下：



还要注意：虽然调整成了1000但是测试无法完成，原因是，还要和当前服务器的环境配置有关系，并不是说调整成1000，就能支持到1000.



5、一般在linux系统下面，apache的mpm方式是预派生模式。有一个推荐的配置。

**在linux下一般是perfor模式**

**给大家一个合理的建议配置. 对大部分网站，中型网站，配置:**

**<IfModule mpm\_prefork\_module>  
              StartServers         5 #预先启动  
              MinSpareServers      5  
              MaxSpareServers      10 #最大空闲进程  
              ServerLimit          1500 #用于修改apache编程参数  
              MaxClients           1000 #最大并发数  
              MaxRequestsPerChild  0 #一个进程对应的线程数，对worker 更有效果。如果是0则不让进程死掉。**

**</IfModule>**

**如果你的网站pv值 百万**

**ServerLimit          2500 #用于修改apache编程参数  
MaxClients           2000 #最大并发数**

总结：

1大型网站的标准（pv,uv,独立ip）

2、带来的问题：大并发，大流量，大的存储

3、大并发解决。网站服务器架构调整（负载均衡，集群，数据库的读写分离）

4、大流量的解决：防止被盗链，启用压缩，启用缓存，减少http请求，增加带宽。

5、大存储的解决：缓存，表要符合三范式，添加索引，存储过程，对sql语句调优，分表和分区，服务器的配置加强。

6、静态化，真静态，伪静态，

真静态实现方式：ob缓存，模板替换技术

伪静态：正则替换，rewrite机制。

7、ab.exe 工具使用。

ab.exe –n 访问的总的次数 -c并发量 页面地址。

mpm（多路 处理模块）并发的处理方式

常见的有：预派生，工作者，winnt,

如何查看自己的mpm是哪种模式：httpd.exe –l