性能测试实战经典案例分享：一个你不知道的压力测试工具

　　在项目上线之前，都需要做[压力测试](javascript:;" \t "_self)，目的是看下我们的网站能抗住多少的压力，能承担多少并发，如果不做压力测试，一旦出现大访问量时，我们的网站会挂掉。

**一、Webbench测试并发**

　　Webbench是[Linux](javascript:;" \t "_self)下的一个网站压力测试工具，能测试处在相同硬件上，不同服务的性能以及不同硬件上同一个服务的运行状况。webbench的标准测试可以向我们展示服务器的两项内容：每分钟相应请求数和每秒钟传输数据量。webbench最多可以模拟3万个并发连接去测试网站的负载能力。

　　测试的环境是 Linux Ubuntu

**1、安装**

**1.1 安装ctags**

　　apt-get install exuberant-ctags

　　ctags 为webbench的依赖

**1.2 下载安装**

　　官网：http://home.tiscali.cz/~cz210...

|  |
| --- |
| root@corwien:~# wget http://home.tiscali.cz/~cz210552/distfiles/webbench-1.5.tar.gz  　　root@corwien:~# tar zxvf webbench-1.5.tar.gz  　　root@corwien:~# cd webbench-1.5/  　　root@corwien:~/webbench-1.5# make  　　root@corwien:~/webbench-1.5# make install  　　root@corwien:~/webbench-1.5# webbench  　　webbench [option]... URL  　　 -f|--force Don't wait for reply from **[server](javascript:;" \t "_self)**.  　　 -r|--reload Send reload request - Pragma: no-cache.  　　 -t|--time <sec> Run benchmark for <sec> seconds. Default 30.  　　 -p|--proxy <server:port> Use proxy server for request.  　　 -c|--clients <n> Run <n> HTTP clients at once. Default one.  　　 -9|--http09 Use HTTP/0.9 style requests.  　　 -1|--http10 Use HTTP/1.0 protocol.  　　 -2|--http11 Use HTTP/1.1 protocol.  　　 --get Use GET request method.  　　 --head Use HEAD request method.  　　 --options Use OPTIONS request method.  　　 --trace Use TRACE request method.  　　 -?|-h|--help This information.  　　 -V|--version Display program version. |

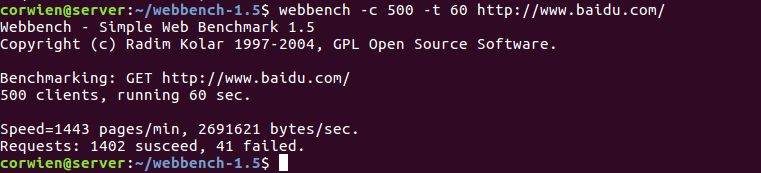
**2、测试**

　　用法：

|  |
| --- |
| // webbench -c 并发数 -t 运行测试时间 URL  　　 webbench -c 100 -t 10 http://baidu.com/ |

　　这里使用[百度](javascript:;" \t "_self)做个试验 ^\_^：

**测试结果：**

[](http://www.51testing.com/batch.download.php?aid=75519)

**结果分析：**

　　每秒钟响应请求数:1443/60= X pages/sec，每秒钟传输数据量2691621 bytes/sec。

　　当并发500时，成功请求1402个，已经显示有41个连接failed了，说明超负荷了。

**3、小结：**

　　1、压力及[性能测试](javascript:;" \t "_self)工作应该放到产品上线之前，而不是上线以后；

　　2、测试时并发应当由小逐渐加大，比如并发100时观察一下网站负载是多少、打开页面是否流畅，并发200时又是多少、网站打开缓慢时并发是多少、网站打不开时并发又是多少；

　　3、更详细的进行某个页面测试，如电商网站可以着重测试购物车、推广页面等，因为这些页面占整个网站访问量比重较大。

　　备注：webbench 做压力及性能测试时，该软件自身也会消耗CPU和内存资源，为了测试准确，建议将 webbench 安装在其他的服务器上，已达到测试数据更加精确。

**二、实战**

　　上边学习了怎样使用webbench来做压力测试，现在就用这个工具来测试下自己的博客，我的博客服务器使用的是阿里云ECS，当并发由100 到 500时，看下服务器的CPU使用率和内存使用情况，当并发数过多时，CPU会不会被占用完，网站此时还能否正常访问，我们的目的就是测出网站能抗住多少的并发量。

**1、使用 top 命令查看服务器资源使用情况**

　　在实测之前，首先学下top命令的参数含义：

　　top命令是Linux下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况，类似于[Windows](javascript:;" \t "_self)的任务管理器。

　　top显示系统当前的进程和其他状况,是一个动态显示过程,即可以通过用户按键来不断刷新当前状态.如果在前台执行该命令,它将独占前台,直到用户终止该程序为止. 比较准确的说,top命令提供了实时的对系统处理器的状态监视.它将显示系统中CPU最“敏感”的任务列表.该命令可以按CPU使用.内存使用和执行时间对任务进行排序；而且该命令的很多特性都可以通过交互式命令或者在个人定制文件中进行设定.

|  |
| --- |
| root@hey:~# top -d 2  　　top - 01:22:59 up 690 days,  9:42,  1 user,  load average: 0.09, 0.05, 0.05  　　Tasks: 117 total,   2 running, 115 sleeping,   0 stopped,   0 zombie  　　%Cpu(s):  0.0 us,  0.5 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.5 st  　　KiB Mem:   1016272 total,   886640 used,   129632 free,   163252 buffers  　　KiB Swap:  1048572 total,    37120 used,  1011452 free.   449744 cached Mem  　　  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND  　　15875 root      20   0  139156  15048   9420 S  0.5  1.5  15:17.66 AliYunDun  　　    1 root      20   0   33372   1388    320 S  0.0  0.1   0:21.49 init  　　    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd |

　　统计信息区前五行是系统整体的统计信息。第一行是任务队列信息，同 uptime 命令的执行结果。其内容如下：

　　01:22:59 当前时间

　　up 690 days,  9:42, 系统运行时间，格式为 天，时:分

　　1 user,  当前登录用户数

　　load average: 0.09, 0.05, 0.05 系统负载，即任务队列的平均长度。三个数值分别为 1分钟、5分钟、15分钟前到现在的平均值。

　　第二、三行为进程和CPU的信息。当有多个CPU时，这些内容可能会超过两行。内容如下：

|  |
| --- |
| total 进程总数  　　running 正在运行的进程数  　　sleeping 睡眠的进程数  　　stopped 停止的进程数  　　zombie 僵尸进程数  　　Cpu(s):  　　0.3% us 用户空间占用CPU百分比  　　1.0% sy 内核空间占用CPU百分比  　　0.0% ni 用户进程空间内改变过优先级的进程占用CPU百分比  　　98.7% id 空闲CPU百分比  　　0.0% wa 等待输入输出的CPU时间百分比  　　0.0%hi：硬件CPU中断占用百分比  　　0.0%si：软中断占用百分比  　　0.0%st：虚拟机占用百分比 |

　　最后两行为内存信息。内容如下：

|  |
| --- |
| Mem:  　　191272k total    物理内存总量  　　173656k used    使用的物理内存总量  　　17616k free    空闲内存总量  　　22052k buffers    用作内核缓存的内存量  　　Swap:  　　192772k total    交换区总量  　　0k used    使用的交换区总量  　　192772k free    空闲交换区总量  　　123988k cached    缓冲的交换区总量,内存中的内容被换出到交换区，而后又被换入到内存，但使用过的交换区尚未被覆盖，该数值即为这些内容已存在于内存中的交换区的大小,相应的内存再次被换出时可不必再对交换区写入。 |

　　进程信息区统计信息区域的下方显示了各个进程的详细信息。首先来认识一下各列的含义。

|  |
| --- |
| 序号  列名    含义  　　a    PID     进程id  　　b    PPID    父进程id  　　c    RUSER   Real user name  　　d    UID     进程所有者的用户id  　　e    USER    进程所有者的用户名  　　f    GROUP   进程所有者的组名  　　g    TTY     启动进程的终端名。不是从终端启动的进程则显示为 ?  　　h    PR      优先级  　　i    NI      nice值。负值表示高优先级，正值表示低优先级  　　j    P       最后使用的CPU，仅在多CPU环境下有意义  　　k    %CPU    上次更新到现在的CPU时间占用百分比  　　l    TIME    进程使用的CPU时间总计，单位秒  　　m    TIME+   进程使用的CPU时间总计，单位1/100秒  　　n    %MEM    进程使用的物理内存百分比  　　o    VIRT    进程使用的虚拟内存总量，单位kb。VIRT=SWAP+RES  　　p    SWAP    进程使用的虚拟内存中，被换出的大小，单位kb。  　　q    RES     进程使用的、未被换出的物理内存大小，单位kb。RES=CODE+DATA  　　r    CODE    可执行代码占用的物理内存大小，单位kb  　　s    DATA    可执行代码以外的部分(数据段+栈)占用的物理内存大小，单位kb  　　t    SHR     共享内存大小，单位kb  　　u    nFLT    页面错误次数  　　v    nDRT    最后一次写入到现在，被修改过的页面数。  　　w    S       进程状态(D=不可中断的睡眠状态,R=运行,S=睡眠,T=跟踪/停止,Z=僵尸进程)  　　x    COMMAND 命令名/命令行  　　y    WCHAN   若该进程在睡眠，则显示睡眠中的系统函数名  　　z    Flags   任务标志，参考 sched.h |

　　默认情况下仅显示比较重要的 PID、USER、PR、NI、VIRT、RES、SHR、S、%CPU、%MEM、TIME+、COMMAND 列。可以通过下面的快捷键来更改显示内容。

　　更改显示内容通过 f 键可以选择显示的内容。按 f 键之后会显示列的列表，按 a-z 即可显示或隐藏对应的列，最后按回车键确定。

　　按 o 键可以改变列的显示顺序。按小写的 a-z 可以将相应的列向右移动，而大写的 A-Z 可以将相应的列向左移动。最后按回车键确定。

　　按大写的 F 或 O 键，然后按 a-z 可以将进程按照相应的列进行排序。而大写的 R 键可以将当前的排序倒转。

**命令使用**

　　top使用格式

　　top [-] [d] [p] [q] [c] [C] [S] [s] [n]

　　参数说明

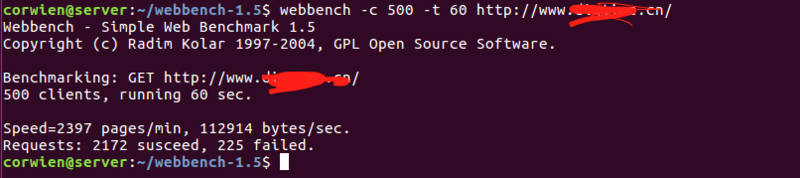
|  |
| --- |
| d 指定每两次屏幕信息刷新之间的时间间隔。当然用户可以使用s交互命令来改变之。  　　p 通过指定监控进程ID来仅仅监控某个进程的状态。  　　q 该选项将使top没有任何延迟的进行刷新。如果调用程序有超级用户权限，那么top将以尽可能高的优先级运行。  　　S 指定累计模式  　　s 使top命令在安全模式中运行。这将去除交互命令所带来的潜在危险。  　　i 使top不显示任何闲置或者僵死进程。  　　c 显示整个命令行而不只是显示命令名 |

**2、压测并同时查看服务器top资源使用情况**

**1、500并发量压测**

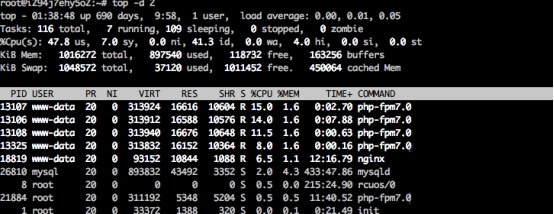
　　root@corwien:~# webbench -c 500 -t 60 http://myblog.com/index.php

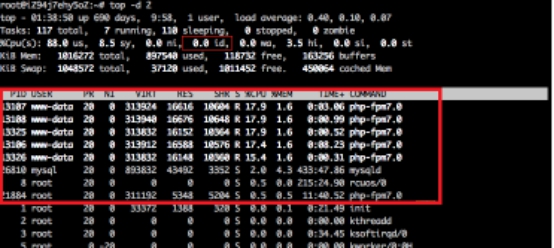
**压测结果：**

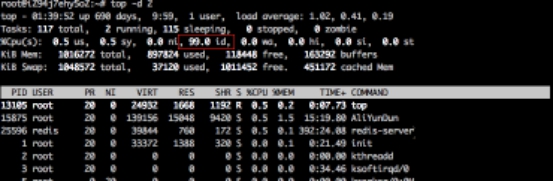
[](http://www.51testing.com/batch.download.php?aid=75520)

　　500个并发，在60秒内，请求成功2172个，失败数225个

　　我们再看下在压测时，服务器的资源使用情况：







　　通过上边的三张图，我们可以看到，当500并发压测时，空闲CPU百分比越来越少，由99.0 id 减少到 41.3 id 再到 0.0 id，压测结束时，又恢复到正常的水平，99.0 id。说明我的网站500并发就扛不住了，CPU资源消耗完了，这时如果访问我的网站，会出现 502 的情况。所以，根据压测结果，可以更好的对网站的硬件配置进行提升和对站点的静态优化。