## 工具

### ab(apachebench)

ab –c100 –n20000 [www.baidu.com/](http://www.baidu.com/)

-c:并发用户数

-n:请求数

-k 使用持久连接

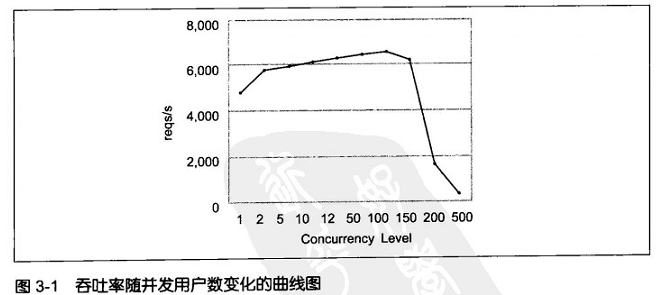
测试核心数据：

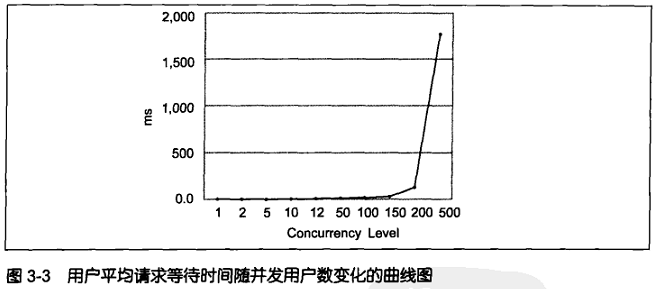
吞吐率：每秒完成的请求数

请求处理时间

请求等待时间







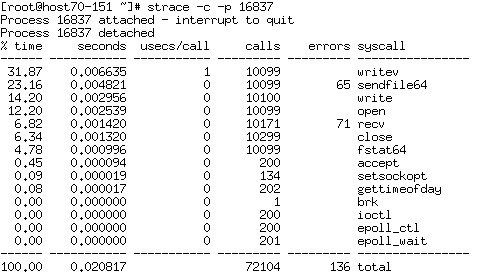
### 测试实例

我们对HTTP长连接做一个测试踊跃

ab -k -c100 -n10000 <http://dev.jizhiinfo.net/test.txt>

Time taken for tests: 3.214 seconds

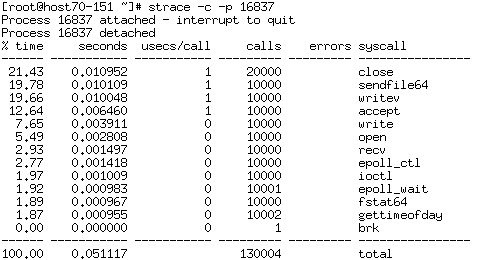
Requests per second: 3111.13 [#/sec] (mean)



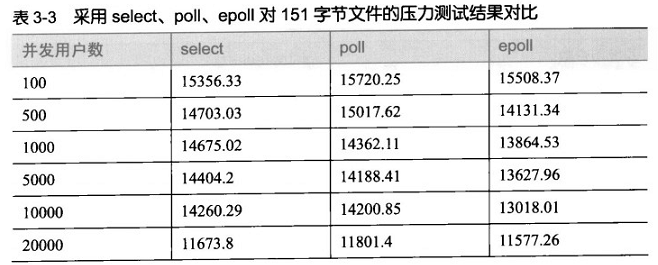
./ab -c100 -n10000 <http://dev.jizhiinfo.net/test.txt>

Time taken for tests: 6.073 seconds

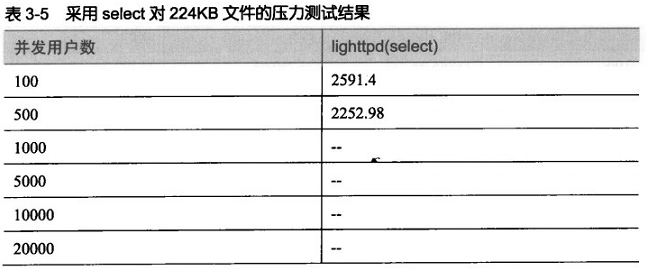
Requests per second: 1646.72 [#/sec] (mean)



不同ＩＯ模型的一个测试数据：



SELECT的IO模型存在１０２４连接数的上限问题，下面又是另外一个结果：



### strace/ltrace

-T 统计每一个系统调用所花费的时间

-c汇总统计系统调用次数，占用的时间比

-tt 时间点打印出来，精确到微秒

strace –T ./tt

strace –c ./tt

### httpwatch

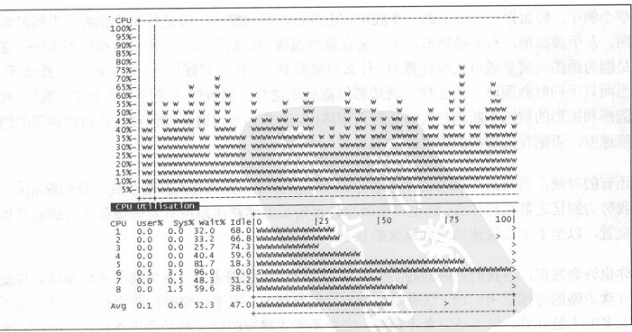
面现前端的HTTP跟踪．

### Free/top/vmstat

运行时进程队列长度:top

load average: 0.00, 0.00, 0.04分别记录，队列在最近１分钟，５分钟１５分钟平均的长度．

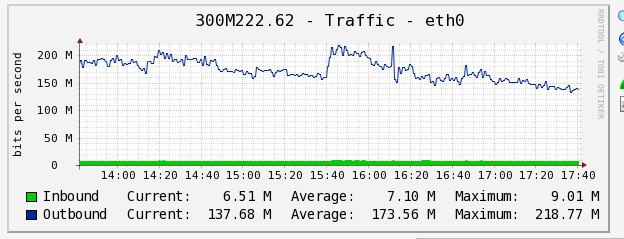
### Nmon



### Cacti／Nagios

cacti基本SNMP协议，对服务器进行监控管理．

如带宽流量



## 如何高性能?

部署多层级的缓存(Cache)，提升网站的处理能力．

### 动态内容缓存

Nginx对于静态文件处理非常高效，每秒可以完成15K的用户请求．但如果我们将请求送到后端的应用服务器，效率将大大下降，如之前我们对一些没有优化的应用做测试数据：60req/S，这是无法接受的一个事实．

思路就是将View这一层的数据，直接缓存起来．

比如：整个页面，数据库连接池

### 动态脚本加速(OPCODE 缓存)

我们使用的语言都是高级语言，都是机器不能识别的语言，如果要让机器执行我们的程序，需要将代码翻译成机器语言，这个翻译的过程，有两种方式，一种是编译过程(compile)，一种是解释过程(parse)．

编译性语言：

我们的C语言就是编译性语言，它在编译的时候就将程序转成了可执行的机器语言．

解释性语言：

php, perl, python, ruby这类我们称之为解释性语言，事先不用做编译，在程序执行时做语言的翻译．这里以PHP为例，PHP的Zend Engine解释器核心引擎，它也是看不懂脚本语言的，在脚本语言与机器语言之间会生成一个中间代码，解释器引擎可以读懂它并执行，这个中间代码，我们也称它为操作码(Operate Code, opcode)．PHP是构建在Zend虚拟机(Zend VM)之上的。Opcode也就是Zend虚拟机中的指令。

Opcode这个过程会做什么呢：语法分析，语义分析等，也类似于编译器的工作了．每一次解释器引擎读入php文件，都会对它进行中间代码的操作．opcode还有另一种称谓: 字节码(byte codes)。 例如Java虚拟机(JVM)，.NET的通用中间语言(CIL: Common Intermeditate Language)等等。

java很特殊，java程序也需要编译，但是没有直接编译称为机器语言，而是编译称为字节码，然后用解释方式执行字节码。因为它还有一个虚拟机．

脚本语言在我们工作经常会碰到，在书中：P91:perl的空闲连接测试脚本，或者mysql的安装脚本mysql/sql-bench/bench-init.pl．它的优势就是开发高效．

如网易的网游的架构就是C/C++和lua脚本相结合．

Zend(它原由仍在以色列技术学院的学生Andi Gutmans 与 Zeev Suraski所开发。他们之后在以色列的 Ramat Gan 创立了 Zend技术公司。Zend 一名为他们名字Zeev和Andi所组成的新字。)

[root@host70-151 bin]# ./php -v

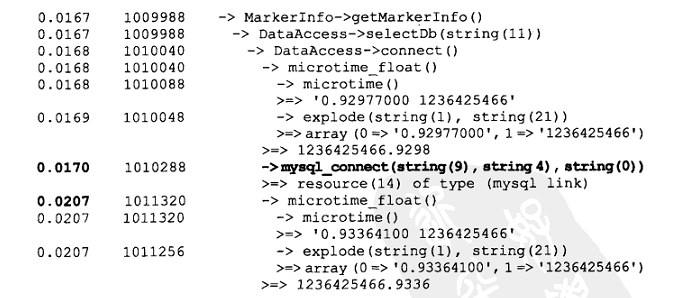
PHP 5.2.17 (cli) (built: Aug 31 2011 18:14:41)

Copyright (c) 1997-2010 The PHP Group

Zend Engine v2.2.0, Copyright (c) 1998-2010 Zend Technologies

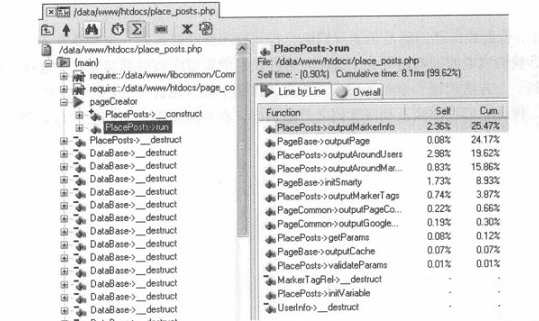
目前PHP的opcode缓存的选择比较多，主流的为：eaccelerator，ZendOptimizer．

另外一个重点：脚本跟踪Xdebug，它深入到跟踪每个解释执行过程:



如上就可以清楚看到，数据库的连接占去了大部分的执行时间．

同时它也可以以图表方式对程序进行分析，如哪个节点占用的时间最多．



### 浏览器缓存

浏览器的三大排版引擎：Trident，Gecko，WebKit

IE缓存写文件，firefox会有一部分写在内存．

HTTP1.1下主流浏览器在单个主机下的并发连接数：  
IE6                                      2  
IE7                                      2  
IE8                                      6  
Firefox2                               2  
Firefox3                               6  
Safari 3,4                      4  
Chrome 1,2                    6   
Opera 9.63,10.00alpha     4

Expire/Cach-control 直接就不访问了

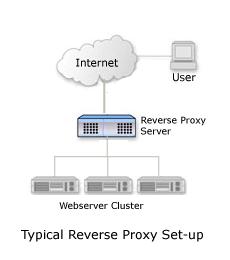
ETag/Last-modify 304不下载内容

### WEB服务器缓存

有些WEB服务器是带有缓存的，如：apache

如Ncache，新浪开源的一个小项目，一个文件，代码２.５Ｋ行，以nginx module的形式加载到nginx服务器上．

### 反向代理缓存



反向代理服务器基于HTTP应用层实现，

当用户浏览器发出一个HTTP请求时，通过域名解析将请求定向到反向代理服务器（如果要实现多个WEB 服务器的反向代理，需要将多个WEB服务器的域名都指向反向代理服务器）。由反向代理服务器处理器请求。反向代理会缓存可缓冲的数据（比如html网 页和图片等）。它根据从WEB服务器返回的HTTP头标记来缓冲静态页面。有四个最重要HTTP头标 记：

Last-Modified: 告诉反向代理页面什么时间被修改

Expires: 告诉反向代理页面什么时间应该从缓冲区中删除

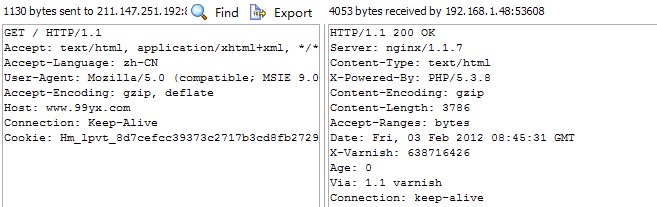
Cache-Control: 告诉反向代理页面是否应该被缓冲

Pragma: 告诉反向代理页面是否应该被缓冲.

例如：在默认情况下，动态页面返回” Cache-control: private.” ，所以动态页面时不会在反向代理服务器缓存的

常见的反向代理服务器：varnish，squid

Varnish是一款高性能的开源HTTP加速器，挪威最大的在线报纸 Verdens Gang 使用3台Varnish代替了原来的12台Squid，性能比以前更好。



### 分布式缓存

目前最为广泛使用的就是Memcached和redis

### 测速工具

<http://www.webkaka.com/webCheck.aspx>

<http://www.jiankongbao.com/labs/http>