

文章编号:1001-9081(2005)06-1458-03

Web Server 应答负载的生成策略

霍莉萍, 郭成城

(武汉大学 电子信息学院, 湖北 武汉 430079)

(icehuo@163.com)

摘要:在分析 Web 服务器应答负载特性的基础上,提出了一种服务器应答负载文件的生成策略。所生成的服务器应答负载文件能满足客户的请求,适用于 Web 服务器的测试研究。

关键词:服务器性能评价; 应答负载特性; 生成策略

中图分类号: TP339 **文献标识码:** A

Strategy of generating Web Server's responding workload

HUO Li-ping, GUO Cheng-cheng

(School of Electronic & Information, Wuhan University, Wuhan Hubei 430079, China)

Abstract: The characterizing analysis of workload generated by Web server's responding is important for Web server performance evaluation. Characteristic of Web server's responding and proper workload help for understanding how server and networks respond to variation load. This paper referred to a strategy of generating responding workload file of Web Server.

Key words: server performance evaluation; characteristic of responding workload; strategy of generation

0 引言

随着服务器端处理任务的日益复杂以及网站访问量的迅速增长,服务器性能的优化就成了非常迫切的任务。在优化之前,最好能确定影响服务器性能的关键因素及限制服务器整体性能的瓶颈,这可以通过测试的方式来实现。测试软件由两部分组成,一部分是用于向被测试的 Web Server 产生请求的客户负载,另一部分是放在被测试的 Web Server 上的供客户访问的文件集,即应答负载。现有的测试软件有 NetBench、WebBench、SPECWeb99 等,但是这些测试软件存在着一些问题:如 SPECWeb99 中只有几种供客户访问的固定长度的文件^[1],WebBench 中只提供了一个固定的供客户访问的 Script 脚本^[2]。这种应答负载分布不能模拟近几年的客户请求特性。

本文在分析 Web Server 应答负载特性的基础上,基于本实验室的实验环境,生成 Web Server 响应客户请求所需的应答负载。

1 应答负载特性分析

客户的请求主要包含两类,一类是读取特定的文本文档或者是图像,这些都属于静态请求的范畴;另一类是执行服务器端指定的 Script 程序,这属于动态请求的范畴。这两类请求占了客户请求的 99%,所以我们主要讨论服务器处理这两类请求所产生的应答负载。

1.1 静态应答负载特性

我们用一些热门网站的日志文件^[3]作为分析对象。对于这些被访问的文件,先对其文件大小进行对数变换,这样可以对大范围的文件大小进行分析。访问负载的分布特性如图 1 所示,从中可以看出,大部分的文件大小处于 100Byte 与 100KByte 之间,且服从以 μ, σ 为参数的对数正态分布。其概

率密度函数为 $f_1(s) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\log S - \mu)^2}{2\sigma^2}}$, 分布函数为 $F_1(s) =$

$\frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\log S} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$, 其中 S 为文件大小,单位为 Byte。

虽然大部分文件小于 100KByte,也存在一些文件,它们相对大得多。对文件大小分布特性的尾部进行分析时不再考虑那些较小的文件的行为。其分布特性如图 2 所示。

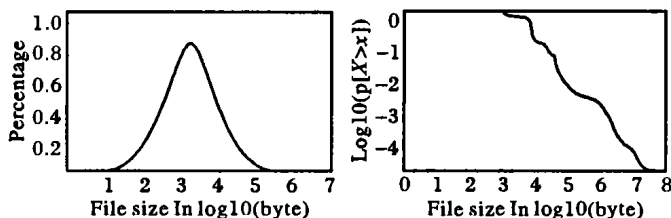


图1 静态文件大小的分布特性

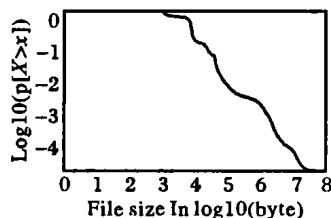


图2 文件分布的尾部特性

从图 2 可以看出文件分布特性曲线的尾部呈现出一种线性行为,这说明文件大小的分布呈重尾性^[4]。服从以 k, α 为参数的 Pareto 分布,其概率密度函数为 $f_2(s) = \frac{\alpha k^\alpha}{s^{\alpha+1}}$, 分布函数为 $F_2(s) = 1 - (k/s)^\alpha$ 。

以上结果表明静态文件服从一种前部为对数正态分布、尾部为 Pareto 分布的混合分布^[5], 客户访问特别小(小于 500Byte)和访问较大(大于 40KByte)文件的次数都很少。

1.2 动态应答负载特性

Web Server 接收到动态请求,先将指定的 Script 程序加载到内存,然后执行该 Script 程序,最后将执行后生成的页面返回给客户。现在有一些网站,它们的网页内容基本上存储在一个数据库中,当有页面请求时,这些独立的内容才被从数据库中查询出来组合成一个网页回送给客户。已有的研究结果表明 Web Server 处理动态请求的平均时间为 0.1s^[6],而对于需要查询数据库的动态请求,处理时间相对较长,一般可以达

收稿日期:2004-11-15;修订日期:2005-02-21

作者简介:霍莉萍(1981-)女,河南项城人,硕士研究生,主要研究方向:信息网络服务器集群系统; 郭成城(1961-),男,北京人,副教授,博士研究生,主要研究方向:信息网络服务器集群系统。

到几秒甚至十几秒。生成的动态页面的大小可以用负指数分布来模拟,其分布函数^[7]为 $F(s) = 1 - e^{-\lambda s}$ 。令 $E(S) = s = \frac{1}{\lambda}$, 则由 $S = -s \log^{(1-\xi)}$ 得到的文件长度 S 符合参数为 λ 的负指数分布,其中 ξ 为在 $[0,1]$ 间均匀分配的随机数。可证明如下: $P(S \leq s) = P(-\frac{1}{\lambda} \log^{(1-\xi)} \leq s) = P(\xi \leq 1 - e^{-\lambda s}) = 1 - e^{-\lambda s}$ 。理论分析和我们的仿真试验结果表明 Script 程序的执行时间也可近似为负指数分布,则可得 $T = -\log^{(1-\xi)}$, 其中 $t = E(T) = 0.1s$ 。

2 文件分配

2.1 静态负载

设对数正态分布和重尾分布的分界点处的文件的大小为 M byte。若小于 M 的文件集占总文件数的比例为 q , 则大于 M 的文件集所占比例为 $1 - q$ 。设文件总数为 z , 则两类文件集的文件数分别为 $q * z, (1 - q) * z$ 。

每一文件集中按文件长度合理分成 a, b, c, \dots 几类, 设每一类的概率为 $P_i(m, n)$, 则 $P_i(m, n) = F_j(n) - F_j(m)$, 其中 $i = a, b, c, \dots, j = 1, 2$ 。对于对数正态分布, m, n 分别为各个类的起始文件大小和最后一个文件大小的对数值; 对于 Pareto 分布, m, n 分别为各个类的起始文件大小和最后一个文件大小。因此各个类占其所属的文件集的比例为: $t_i = P_i / (P_a + P_b + P_c)$ 。最后分别计算出两个区域内的每一类文件占整个区域文件的比例, 各个类中再按照文件大小均匀分配文件数。

2.2 动态负载

通过实时监测 Script 程序的运行时间, 按客户的动态请求(在我们实验环境下, 该请求包含随机产生的要求动态脚本执行的时间)进行相应时间长度的请求处理, 并将处理结果返回给客户。

3 应答负载文件生成算法

静态负载的生成算法用 VC 编程实现, 每一类文件中以轮转的方式均匀分配文件数; 对于动态负载, 由于 PHP 语言在控制程序执行时间方面比较有优越性, 所以其生成算法用 PHP 语言编程实现。

1) 静态负载生成算法描述

```
for(i = m, i <= n, i++) /* m: 某类文件中文件名的起 */
{
    /* 始值, 文件以 i.html 命名 */
    Generate();
}
Generate 函数:
/* 在同一文件类中按文件大小均匀分配文件个数 */
{
    fwrite(); /* 以字母 a 来写入 html 文件, 用其个数 */
    /* 来代表文件的长度 */
    ByteNum = Mix + k * diff;
    /* 计算当前文件的大小, 同一文件类中文件大小以 diff 递 */
    /* 增, 根据文件类的流行程度来确定 diff 的值 */
    k++;
    if (ByteNum >= Max) /* Max 为该文件类中最大文件的大小 */
    {
        k = 1; /* k 置 1 后重新进入循环, 再次产生 */
        /* 一轮具有同样大小特性的文件, */
    }
}
```

2) 动态负载生成算法描述

```
<?php
starttime = getmicrotime();
```

```
/* 获取程序开始运行的时间 */
do {
    while( $ count) {
        $ par4 = $ par1 * $ par2;
        $ par4 /= $ par3;
        $ par4 += $ par1;
        $ count - -;
    }
    /* 简单的算术操作以消耗程序运行时间 */
    srand( ( double) microtime() * 100);
    /* 取得乱数种子, 执行时以百分之一的随机率 */
    /* 来产生随机数 */
    $ randval = rand(0, 10);
    /* 在 0 和 10 之间取一个数字 */
    for ( $ i=0; $ i <= $ randval; $ i++) {
        /* 生成页面 */ $ fp = fopen();
        fwrite();
        fclose( $ fp);
    }
    $ endtime = getmicrotime();
    /* 获取程序运行一段时间后的当前时间 */
    $ processtime = $ endtime - $ starttime;
    /* 程序已运行的时间 */
} while( $ processtime < $ T)
/* $ T 为客户随机产生的请求动态脚本的执行时间 */
?>
getmicrotime 函数:
/* 获取当前系统时间的函数 */
{
    list( $ usec, $ sec) = explode(" ", microtime());
    return ((float) $ usec + (float) $ sec);
}
```

4 生成负载验证

4.1 静态负载

取静态负载文件分布特性分界点处文件的大小^[8]为 $M = 10\text{KByte}$, $p = 85\%$, $\mu = 9.357$, $\sigma = 1.318$, $k = 133\text{K}$, $a = 1.1$ 。把符合两种负载分布的文件都分成 3 类, 分别是: $0 - 100\text{Byte}$, $100\text{Byte} - 1\text{KByte}$, $1 - 10\text{KByte}$; $10 - 40\text{KByte}$, $40 - 100\text{KByte}$, $100\text{KByte} - 1\text{MByte}$ 。则生成的各类文件所占的比例如表 1 所示。

表 1 文件按大小分类

文件大小	0 - 100B	100B - 1KB	1 - 10KB	10 - 40KB	40 - 100KB	100KB - 1MB
比例 (%)	1.920	11.458	71.621	11.810	2.087	1.104

文件比例的分布基本上符合上面提到过的混合分布。以长度为 $1 - 10\text{KByte}$ 这类离分界点较近的文件为例, 若我们需要产生包含 1000 个文件的负载文件集, 则 $1 - 10\text{kbyte}$ 的文件应有 716 个。对 Web 访问的大量研究工作表明, 长度为 $1 - 10\text{kbyte}$ 的文件被访问的次数较多^[5], 故应该在该文件类中放置分布较密集的文件, 我们选择该类文件的文件大小递增值 (diff) 为 100Byte , 仿真结果如表 2 所示。

表 2 同一类中不同大小的文件分配

文件大小 (KByte)	1.1	1.2	1.3	...	2.6	2.7	2.8	...	10
个数	8	8	8	8	8	7	7	7	7

从表 2 可以看出, 静态负载每一类文件中不同大小的文

件个数基本上达到了均匀分配。

4.2 动态负载

将上述 PHP Script 程序分别放在两台不同配置的服务器上进行测试。测试结果如图 3 所示,时间单位以 ms 计。

Server1. PIII800, 256M 内存, 18G 硬盘

Server2. PII350, 64M 内存, 8G 硬盘

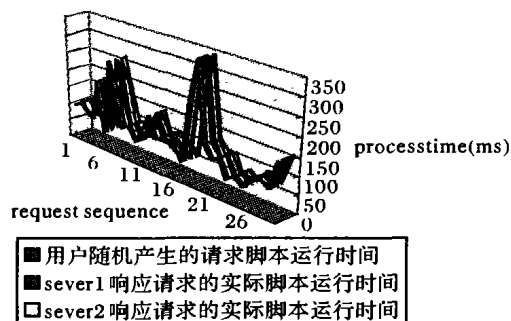


图3 动态负载的实际执行时间

从图3可以看出,服务器能够按照客户随机产生的脚本执行时间请求来控制 Script 程序的执行时间。且对于同一个请求,不同配置的服务器实际脚本运行时间基本上是相同的。与那些通过限制一定的程序循环次数来达到控制执行时间的 Script 程序相比,该 Script 程序的执行时间不随被测试的服务器配置的不同而不同。因为同样的程序循环次数在配置不同的服务器上运行时,由于操作速率的不同,不能保证有相同的执行时间,而上述的 Script 程序直接从监测服务器的系统时间出发,程序执行时间不会受服务器操作速率的影响。

5 结语

本文分析了近几年来 Web 访问中服务器端应答负载的

分布特性,并对其进行仿真。对于那些经常被访问的静态文件,我们可以生成多个备份放在在测试文件集中。文中生成的 Script 程序能很好地适用于任何不同配置的服务器。利用这样的负载文件集,我们对本实验室的服务器进行了评测,并取得了比用现有测试软件(如 Webbench)更好的效果。

参考文献:

- [1] 朱晶. Specweb99-Web 服务器性能测试工具[J]. 计算机与现代化. 2001, 4: 1-6.
- [2] Webbench 5.0 测试软件[DB/OL]. <http://www.veritest.com/benchmarks/webbench/default.asp>, 2004.
- [3] Traces available in the Internet Traffic Archive[DB/OL]. <http://ita.ee.lbl.gov>. 2004-4-17
- [4] CROVELLA M, TAQUU M. Estimating the Heavy Tail index from Scaling Properties[J]. Methodology and Computing in Applied Probability, 1999, 1(1): 55-79.
- [5] BARFORD P, BESTAVROS A, BRADLEY A, et al. Changes in Web Client Access Patterns: Characteristics and Caching Implications[DB/OL]. <http://citeseer.nj.nec.com/barford98changes.html>, 1998.
- [6] 林凡. 集群的可扩展性及其分布式体系结构之十: TCP 粘合技术原理[EB/OL]. http://www-900.ibm.com/developerWorks/cn/linux/cluster/cluster_system/balance/part6/index.shtml. 2003-11-03.
- [7] SHI W, COLLINS E, KARAMCHETI V. Modeling object characteristics of dynamic Web content[A]. Global Telecommunications Conference[C]. 2002. GLOBECOM'02. IEEE, 2002, 3(3): 2220-2224.
- [8] BARFORD P, CROVELLA ME. Generating representative Web workloads for network and server performance evaluation[J]. In Proceedings of Performance '98/SIGMETRICS'98, 1998. 151-160.

(上接第 1457 页)

户类别的特征制定营销策略,还要考虑它具有第 1 类客户类别的特征。

本分类模型划分了不同客户种类,区分了不同客户种类的特征。表 2 对实例结果的 5 类客户的特征进行了分析:

表 2 对实例结果的特征分析

类别	客户数量	类别客户特征分析
I	9	成熟客户: 频繁与企业接触, 且累计交易量远大于其他客户
II	27	主要客户: 与企业接触的时间较近, 购买金额较大, 但接触的次数较少
III	6	新客户: 与企业接触的时间较近, 购买金额和接触的次数都较少
IV	5	衰退客户: 较长时间没有与企业接触, 然而其与企业接触的频率和价值贡献都比较高, 可能是有流失危险的有价值客户
V	3	无价值客户: 较长时间没有与企业接触, 且价值不大甚至无利润的客户

分析结果表明, 由于采用 RFM 客户行为分析指标作为客户分类的指标, 划分的客户类别的特征体现了客户对企业利润的贡献率, 客户的忠诚度和客户流失的可能性。

4 结语

本文所讨论的在 CRM 系统下实现的客户分类模型, 采用 RFM 作为客户的分类指标, 使用模糊聚类最大矩阵元法和模

糊 ISODATA 结合作为客户分类的方法, 成功应用于某乳业集团的 CRM 系统中。实现了客户分类, 为差异化对待客户提供了科学依据和基础, 通过有针对性的制定营销策略, 降低成本, 提高了企业的经济效益。

参考文献:

- [1] 纳德 S 史威福特. 杨东龙, 等译. 客户关系管理[M]. 中国经济出版社, 2000.
- [2] 王扶东, 等. CRM 中客户关系分析评价方法研究[J]. 计算机工程与应用, 2003, 31.
- [3] 赵国庆. 客户关系管理中的客户分类方法研究[J]. 安徽机电学院学报, 2001, 16(4).
- [4] 刘义, 等. 基于购买行为的客户细分方法比较研究[J]. 管理科学, 2003, 16(1).
- [5] BEZDEK JC. Physical interpretation of fuzzy ISODATA[M]. IEEE Trans, Systems Man, Cybern, 1976, SME-6.
- [6] BEZDEK JC. Pattern Recognition with Fuzzy Objective Function Algorithms[M]. New York: Plenum, 1981: 65-93.
- [7] HSU TH. An application of fuzzy clustering in group-positioning analysis[J]. Proc Natl Sci, Councl ROC(C), 2000, 10(2): 157-167.
- [8] 于连生. 模糊聚类法的研究——最大矩阵元原理[J]. 吉林大学自然科学学报, 1982, 4: 106-112.
- [9] 张跃. 模糊数学方法及其应用[M]. 北京: 煤炭工业出版社, 1992.
- [10] 朱剑英. 智能系统非线性数学方法[M]. 武汉: 华中科技大学出版社, 2001.