

文章编号: 1007-9432(2006)S1-0078-04

Web 应用软件性能测试工具的研究

李志峥, 杨社堂

(太原理工大学 计算机与软件学院, 山西 太原 030024)

摘 要: 从产生压力、收集数据、分析数据三方面构建了 Web 性能测试工具的基本框架, 并论述了其各部分所采用的技术要点。另外在对现有 Web 性能测试指标以及对 HTTP 协议不同版本的分析和研究的基础上提出了一个新的测试指标——加权响应时间。加权响应时间与连接建立时间和响应时间相比更能体现用户的感受。通过实验证明, 该工具比其他测试工具功能更强大, 且由于我们在分析器中引入了美国加利福尼亚大学 Lawrence Berkeley 国家实验室的 Vern Paxson 提出的一套系统的分析计算理论, 使得最终的测试结果更加精确。

关键词: Web; 性能测试; 虚拟用户; 负载; 控制器

中图分类号: TP311.56 **文献标识码:** A

性能测试的主要手段是通过产生模拟真实业务的压力对被测系统进行加压, 研究被测系统在不同压力情况下的表现, 找出其潜在的瓶颈。因此要对 Web 应用软件进行性能评估的测试工作应该包含三方面的内容。产生压力; 能够对后台系统进行监控, 并收集相关数据; 对压力数据能够进行分析, 快速找出被测系统的瓶颈。本文主要内容是从这三方面构建了测试工具的基本框架并论述了其各部分所采用的技术要点。

1 Web 的基本工作原理^[1]

在 web 环境下, 用户通过浏览器向分布在网络上的许多服务器发出请求, 服务器对浏览器的请求进行处理, 将用户所需信息返回到浏览器。此通信过程采用的是超文本传输协议(HTTP), Web 采用的是 B/S 体系结构的三层结构, 它把二层 C/S 结构的事务处理逻辑模块从客户机的任务中分离出来, 由 Web 服务器单独组成一层来负担其任务, 这样客户机的压力减轻了, 把负荷分配给了 Web 服务器。如图 1 所示。

2 性能测试指标

性能测试需要分析的典型性能度量有^[2]: 性能与用户数量的关系; 性能与时间的关系; 每秒的点击数(每秒的事务或请求); 每个用户交互中的错误数

目; 单位时间中发生的错误数目; 每秒的千字节数目; 每用户下载的平均数据量。

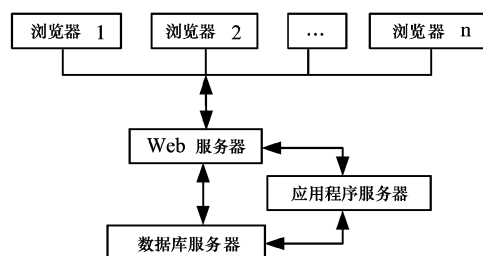


图 1 WEB 工作原理图

由以上给出的内容可知 Web 性能测试的性能指标主要由以下 4 部分组成:

- 1) 最大顺畅连接数。指在满足用户可接受的相应时间或数据流量的前提下, Web 所能支持的最大并发连接数。SpecWeb99 认为如果客户和服务器的连接对各种请求的平均带宽大于 320 kb/s, 则称之为“顺畅连接”。
- 2) 连接建立时间。指从发送连接请求到成功建立连接的时间。
- 3) 相应时间。指发送 HTTP 请求到接收完响应的时间(指标 2, 3 是以一定负载强度为前提)。
- 4) 数据流量。单位时间内传输的字节数。数据流量通常只有和其它测试指标结合起来才是有意义的, 因为传出单个的大文件很容易达到较高的流量, 但通过大量的并发连接传输很多小文件发送请

* 收稿日期: 2006-02-26

作者简介: 李志峥(1977-), 女, 山西太原人, 在读硕士, 主要从事分布式环境下的数据研究, (Tel) 13453192508, (E-mail) jsj-122@126.com

通讯联系人: 杨社堂, 男, 教授, (Tel) 13903403225, (E-mail) STYang@1618.com.cn

求, web 服务器处理请求返回响应这样一个过程。

本文综合连接建立时间和响应时间, 提出一个新的测试指标。以获取页面这样一个事务为例, 每个页面通常包括主体文件和它的一些内嵌对象小文件。用户在浏览一个页面时, 感受最明显的是一个页面的整体传输时间。

假设: 一个连接的建立时间为 S , 一个页面 P 包括 n 个文件, 第 i 个文件的响应时间为 R_i ; T_0 和 T_1 分别为使用 HTTP/1.0 和 HTTP/1.1 获得页面 P 的总时间。则:

$$T_0 = (S + R_1) + (S + R_2) + \dots + (S + R_n) = nS + \sum R_i;$$

$$T_1 = S + R_1 + R_2 \dots R_n = S + \sum R_i.$$

令 R 表示各个文件响应时间的平均值, 则使用 HTTP/1.0 和 HTTP/1.1 获得一个件平均消耗的时间和, 分别为:

$$\bar{T}_0 = \frac{T_0}{n} = S + \frac{1}{n} \sum R_i = S + \bar{R};$$

$$\bar{T}_1 = \frac{T_1}{n} = \frac{1}{n} (S + \sum R_i) = \frac{1}{n} S + \bar{R}.$$

由此, 提出新的测试指标— 加权响应时间, 即: $T = \phi S + R$ 这里, ϕ 为通过一个连接所传送文件数目的倒数(即通过一个连接传送的各个文件分担建立该连接的开销)。加权响应时间与连接建立时间和响应时间相比, 更能体现用户的感受。

3 性能测试工具的体系构架

结合日常使用和熟悉的测试软件^[3], 我们设计出的软件测试工具的基本框架如图 2 所示。

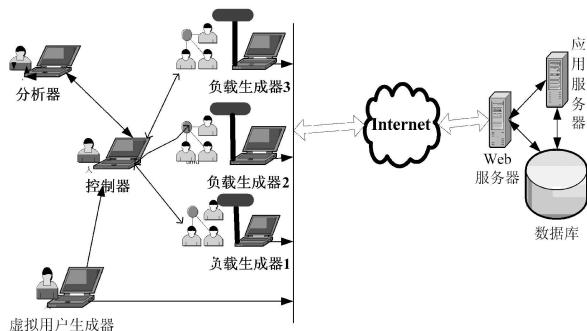


图 2 测试系统框架

由图可知该框架由四部分组成: 虚拟用户生成器; 控制器; 负载生成器; 数据分析器。

3.1 虚拟用户生成器

虚拟用户产生器通过录制客户端和后台服务器之间的通讯包, 分析其中的协议, 自动产生脚本。用户在自动产生的脚本的基础上进行修改, 从而快速

开发出一个逻辑功能和客户端软件完全一样的压力脚本程序。录制的技术主要是模仿了代理服务的方式来实现的。

虚拟用户生成器根据对捕获的数据的分析, 将其还原成对应协议的 API 组成的脚本。虚拟用户生成器的主要的技术要点是如何根据捕获的数据包来反解析成对应的网络协议。通常捕获的数据包为 TCP 数据流, 我们可以很容易的生成 socket 层次的脚本, 类似如下示例:

```
int main( int argc, char * * argv)
{
    char buf[ BUF_MAX_LEN];
    int socket = 0;
    socket = connect(“ IP= 192. 168. 3. 123”, “Port=
3200”, TCP);
    getbuffer( buf, “ datapack_catched”, 1, SEND);
    send( socket, buf);
    receive( socket, buf);
    getbuffer( buf, “ datapack_catched”, 3, SEND);
    send( socket, buf);
    receive( socket, buf);
    ...
    close( socket);
}
```

其中 datapack_catched 包含着录制时捕获的数据包, 按照“发-收-发-收-发-收”的顺序排放。

毫无疑问, 这样的脚本按照记录的收发过程来回放, 但是它的最大的缺点是处于底层。要分析和修改 socket API 的脚本以及数据包的具体内容将是一个繁重而且烦人的工作, 进行关联的工作的难度也将大大提高。客户端程序往往是利用更高层的应用协议 API 编写的, 客户端软件的编写者也不一定对 socket API 组成脚本进行修改。因此, 虚拟用户生成器应该尽可能地产生更高层网络协议的脚本, 方便用户的阅读和修改工作。

3.2 关联问题

客户端和服务端之间的通讯, 有一部分是数据是动态的, 每次通讯都不一样。代理录制器在录制的时候是无法区分哪些是静态的信息, 哪些是动态的信息, 所有的信息都以 hard coded 的方式记录下来。但是在回放的时候, 如果有些信息不改变, 那么脚本是不能执行成功的。考虑如下情形: 如图 3 所示, 用户以 mary/123456 的账号/口令登录某一 Web 服务器, 查询某产品的信息, 由虚拟用户生成器录制交易的全部通讯包。Web 服务器返回给用

户 mary 一个 sessionid: 001, 然后用户再用这个 sessionid: 001 向服务器发出查询产品信息请求, 然后当服务器识别了正确的 sessionid 后, 向用户返回相应的产品信息。

由图 3, 图 4 可知, 当 sessionid 是以 harding code 的形式存在的, 回放测试脚本时就会出错, 这种问题在任何系统都是非常常见的问题。解决这个问题通用的模式是: “服务器返回给客户端一些动态变化的值, 客户端使用这些值去访问服务器的时候, 不能把这些值写死在脚本里面, 而应该存放在一个变量里面。” 这就是关联的概念。图 5 显示了通过关联后回放测试脚本的情况, 很显然关联解决了前面出现的问题。

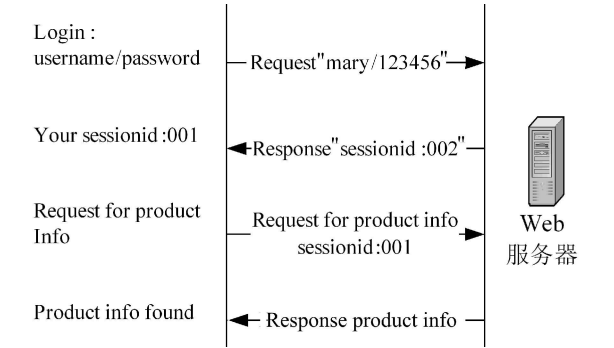


图 3 关联问题示意图 A

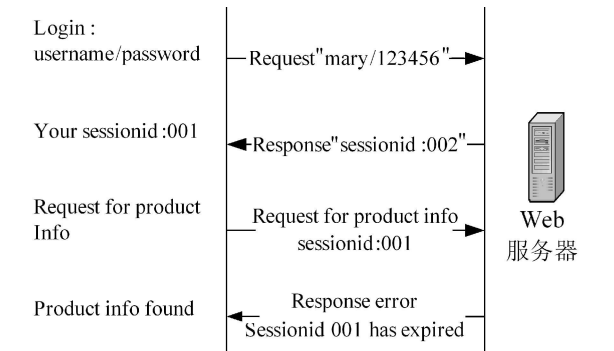


图 4 关联问题示意图 B

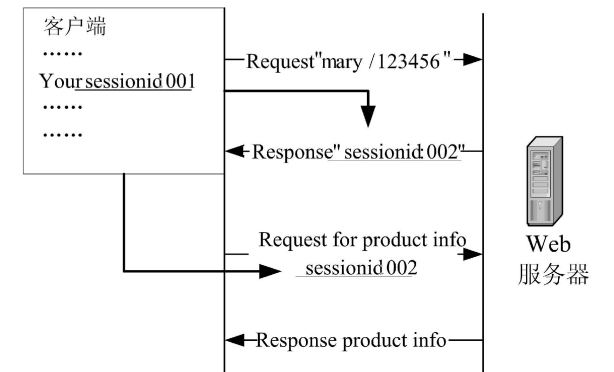


图 5 关联问题示意图 C

关联的工作往往占据开发脚本的大部分时间, 因为我们必需针对每一个具体的系统进行细致的分

析, 确定其需要关联的动态信息。为了快速开发脚本, 虚拟用户生成器必需提供帮助我们关联的手段, 最好做到自动关联。自动关联的方法有三种:

- 1) 在录制之前设定辨别规则, 录制完毕, 产生脚本的时候根据规则识别出需要关联的动态内容, 从而产生正确的脚本。
- 2) 录制完毕回放一遍, 把回放结果与录制结果进行自动对比, 确定动态信息, 进行自动关联。
- 3) 录制两个一模一样的脚本, 对比其中的差异来确定需要关联的动态信息, 然后进行关联。

自动关联的功能是否完整可靠, 关系到我们能否借助虚拟用户生成器快速开发出符合要求的脚本, 因此关联也是虚拟用户生成器中重要的功能。

3.3 控制器和负载生成器部分

控制器是整个性能测试的核心^[4]。负载生成器则是用来产生压力, 它们以进程或者线程的方式运行由虚拟用户生成器生成的测试脚本。负载生成器如何运行脚本, 由控制器来决定。

控制器和负载生成器实际上是一套框架程序。具体执行什么功能, 是由脚本来完成的。控制器和负载生成器的体系结构如图 6 所示。

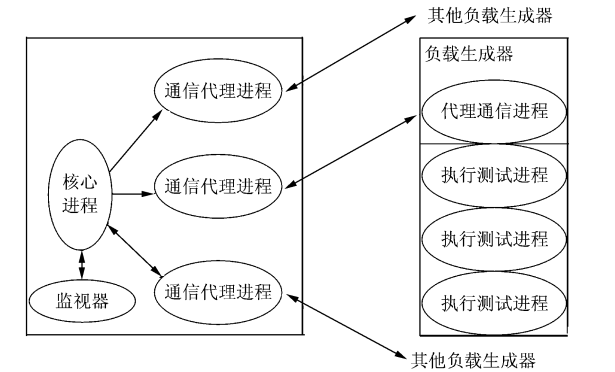


图 6 控制器和负载生成器工作流程图

如图 6 所示, 控制器上面有若干进程/线程。每种进程的作用如下。

核心进程是起整体调度的主进程, 它负责联系和用户界面打交道的工作。

代理通信进程负责和远端的负载生成器中对应的代理通信进程通讯。负责把编译好的脚本传送到负载生成器上。在脚本运行的时候, 定期从 Player 机器上获取 Player 的运行状态。

监视器进程负责对被测试系统的各个环节进行监控, 并把监控的内容一方面写入控制器的本地磁盘, 另外一方面把监控的内容传送给核心进程, 实时地显示在用户界面上。

负载生成器的进程有两种, 一个是代理通信进程, 一个是执行处理进程。执行处理进程负责和控制器通讯, 它根据控制器的指示, 在本机器上派生出指定数目的执行处理线程, 这些执行处理线程负责具体执行相应的脚本。执行处理线程个数就是虚拟用户的个数。

3.4 数据分析器 Analysis 及其技术要点^[9]

数据分析器其实就是一个纯数学工具软件, 它可以根据控制器所给的采用数据进行一系列的数据分析, 并把分析结果以图表的形式反映给用户, 使用户能够更直接的获得测试结果。

作为一个单纯的数学工具来说, 它是很好开发的, 但是在这个模块中我们所要考虑的问题是如何使这个数据分析工具更适合测试数据的分析, 因为

由于环境的影响, 我们所采集的数据并不能全部用来做分析使用, 那么是什么原因影响了这些采样值的准确度, 又如何来剔除这些不符合要求的数据呢? 为了解决这些问题, 我们在软件中采用了美国加利福尼亚大学 Lawrence Berkeley 国家实验室的 Vern Paxson 提出的一套系统的分析计算理论, 由于篇幅的限制, 我们在这里就不详细的做介绍了。

4 总结

我们所设计的这个测试框架包含了测试环节中各方面的内容, 通过编程实现该框架中的功能后, 这个测试工具可以很好的完成 Web 性能测试的工作。从而提高了测试的效率和准确度。

参考文献:

- [1] 古乐, 史九林. 软件测试技术概论[M] . 北京: 清华大学出版社, 2003 06 22
- [2] 王伟. 基于 web 的系统测试方法[DB/O L] . http: //www. vooyoo. com /list /5170. shtml
- [3] 王会艳. 测试工具的选择与使用. www. csdn. net /magazine, http: //www. testage. org /stldio /expert /htm
- [4] Connie U Smith. Performance Solutions A Practical Guide to Creating Responsive Scalable Software[M] . 北京: 机械工业出版社, 2003; 145 200.
- [5] Lydia Ash. The Web Testing Companion[M] . 北京: 机械工业出版社, 2003; 50 80.

Research of Performance Test Tool Based on Web Application Software

LI Zhi zheng, YANG She tang

(College of Computer and Software of TUT, Taiyuan 030024, China)

Abstract: Performance testing process is made of three part: Create stress; collect data and analyze data. The main content of this paper discuss the frame of a testing tool from this three aspect. It has been proved by test that the function of this tool is more powerful than other test tools . The tool is composed of four parts: virtual user generator, controller, loader generator and analyzer. VuGen produce test script automatically by which capture data pack between hosts, technology of capture avail oneself of principle of proxy, controller is center of whole test, loader generators create pressure and analyzer is used to analyze result of test and produce chart or picture.

Key words: Web; performance testing; virtual user; load; control

(编辑: 张爱绒)