

# 基于负载性能指标的 Web 测试

浦云明 王宝玉 (集美大学 计算机工程学院 福建 厦门 361021)

**摘要:** 研究了资源占用率、响应时间、并发用户和思考时间等 Web 应用的性能指标。负载测试是 Web 应用系统测试的一个重要方面,有助于确认被测系统的性能需求和负载增长情况,发现系统的异常。提出了负载测试一般方案,应用 LoadRunner 测试工具,测试了学习交流在线系统,分析了事务、网络、系统响应时间等指标。

**关键词:** Web 应用;负载测试;自动化测试;性能指标;并发

## Web Testing Based on Performance Criteria of Load Testing

PU Yun-Ming, WANG Bao-Yu (School of Computer Engineering, Jimei University, Xiamen 361021, China)

**Abstract:** The Web application performances criteria are studied which include resource occupied, response time, concurrency users and think time. Performance testing is a significant part of the testing for Web applications. Load testing can be used to confirm the performance requirements of the tested system and load growth in the future. It is also used to troubleshoot where the software exceptions happen. With the automatic testing software LoadRunner, an online learning system is tested. The performance of transactions, network and response time and other performance criteria are analyzed.

**Keywords:** Web applications; load testing; automatic testing; performance criteria; concurrency

测试是检验软件开发各阶段成果是否接近预期的目标,尽可能早地发现错误、定位并加以修复。测试能发现引入的错误,也能估计软件错误密度,确保软件满足应用需求及相应的开发标准。在典型的软件开发项目中,软件测试工作量一般占总工作量的 40% 以上。测试工作一般可以划分为静态和动态测试,静态测试包括走查、审查和评审工作,动态测试是编写测试用例和运行程序工作。随着 WEB 应用的增多,测试工程师运行程序、跟踪程序的方法已经无法满足 WEB 应用的测试需求。因此,自动化测试就成为必然选择,自动化测试一般定义为各种测试活动的管理与实施,使用软件来控制测试的执行,完成实际输出和预期输出的比对,测试控制条件的构建,完成测试报告<sup>[1]</sup>。

Web 负载测试的工具主要有 LoadRunner、WebLoad、Etest、OpenSTA、WAS 等,其主要机制是编写或录制测试脚本、运行测试脚本、测试数据分

析和生成测试报告<sup>[2]</sup>。对于 WEB 应用系统的测试,关键是做好性能测试,发现产品性能上的缺陷。负载测试是评价一个 WEB 应用系统的重要手段,负载测试有助于确认被测系统是否能够支持性能需求,是否能够满足预期的负载增长,找出系统出现异常的原因,并对系统性能进行优化<sup>[3]</sup>。

## 1 负载测试性能指标

负载测试是评价一个 Web 应用程序的重要过程,测试系统在某一负载级别上的性能,估算预期的负载增长。负载级别可以是某个时刻同时访问 Web 系统的用户数量,也可以是在线数据处理的数量等。

### 1.1 响应时间

对于 WEB 应用,响应时间是从请求到响应所需时间,从客户端请求开始,结束于来自服务器的响应并呈现页面的时间。从图 1 可以看出,对于一个三层应

基金项目:福建省教育厅基金(JA07133);厦门市科技项目计划(3502Z20083025)

收稿时间:2009-09-15;收到修改稿时间:2009-10-21

用架构, 响应时间包括了网络传输时间( $N1+N2+N3+N4$ )和 Web 服务器和数据库服务器的处理时间( $A1+A2+A3$ )。

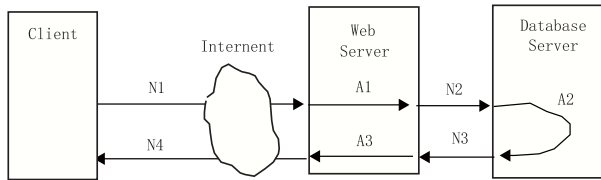


图 1 WEB 应用响应时间分析

对于电子商务网站的响应时间, 在欧洲和美国, 普遍接受的响应时间标准为 2/5/10 秒, 即 2 秒的客户响应时间是非常优秀的, 5 秒的客户响应时间是比较好的, 10 秒是客户能接受的响应时间上限<sup>[4]</sup>。另外一个重要性能指标是 TPS(transaction per second), 即服务器每秒处理的事务数。一个事务是指一个客户机向服务器发送请求, 服务器做出反应的过程。

吞吐率是测试过程中每秒从服务器返回的字节数。从定义上来看, 如果 TPS 很小, 但是吞吐率比较大, 说明服务器返回的页面文件(字节数)是比较大的, 根据页面细分图, 如果页面存在问题, 应考虑页面压缩。如果 A1 或者 A3 比较大, 说明 WEB 服务器处理可能存在问题, 如果 A2 比较大, 则说明 DB 服务器处理存在问题, 建议进行 SQL 优化。

当增大系统的压力(或增加并发用户数)时, 吞吐率和 TPS 的变化曲线呈大体一致, 则系统基本稳定。若压力增大时, 吞吐率的曲线增加到一定程度后出现变化缓慢, 甚至平坦, 同时 TPS 也趋于平坦, 应查看系统资源使用情况, 如果资源使用率比较低, 说明服务器硬件资源不存在问题, 查看网络流量, 估计网络带宽存在问题。同理若点击率和 TPS 曲线出现变化缓慢或者平坦, 如果压力增加时, 点击率趋于平坦, 很可能是服务器响应时间增加, 观察服务器资源使用情况, 确定是否是服务器问题。

评价系统性能均以每秒钟完成的技术交易的数量来衡量。系统整体处理能力取决于处理能力最低模块的 TPS 值。依据经验, 应用系统的处理能力一般要求在 10-100 左右。不同应用系统的 TPS 有着十分大的差别, 一般需要通过性能测试进行准确估算。

## 1.2 并发用户数

一个 OA 系统有 3000 个使用用户, 也就是系统

总用户是 3000 个, 它是“系统用户数”。系统的在线统计发现, 最高峰 800 人在线, 这是同时在线用户数的峰值, 那么系统的并发用户数是多少呢?

并发的第一种情况是所有的用户在同一时刻做同一件事情或者操作, 这种操作一般指做同一类型的业务。比如在信用卡审批业务中, 一定数目的用户在同一时刻对已经完成的审批业务进行提交; 还有一种特例, 即所有用户进行完全一样的操作, 例如在信用卡审批业务中, 所有的用户可以一起申请业务或者修改同一条记录。另外一种并发是广义范围的并发, 这种并发与前一种并发的区别是, 尽管多个用户对系统发出了请求或者进行了操作, 但是这些请求或者操作可以是相同的, 也可以是不同的, 因此, 同一时间进行同一操作的用户并不多, 但对整个系统而言, 仍然有很多用户同时对系统进行不同的操作。

在实际的性能测试中, 一般比较关注业务的并发用户数, 公式(1)和(2)中用来估算并发用户数和峰值<sup>[4]</sup>, 其中 C 是平均的并发用户数, Cp 是并发用户数峰值, n 是 login session 的数量, L 是 login session 的平均长度, login session 定义为用户登录进入系统到退出系统的时间段, 假设用户的 login session 产生符合泊松分布, T 是考察的时间长度<sup>[4]</sup>, 例如一个 OA 系统的考察时间长度为 8 小时。

$$C = n * L / T \quad (1)$$

$$Cp = C + 3 * \sqrt{C} \quad (2)$$

假设 OA 系统有 5000 个用户, 每天访问系统的平均用户数是 800 个, 用户在 8 小时工作时间内使用系统, 一个典型用户, 一天内从登录到退出系统的平均时间为 4 小时, 依据公式(1)和(2)可计算平均并发用户数和峰值用户数。其中,  $C = 800 * 4 / 8 = 400$ ,  $Cp = 400 + 3 * \sqrt{400} = 460$ 。

另外, 关于用户并发的数量, 有 2 种常见的错误观点。一种错误观点是把并发用户数量理解为使用系统的全部用户的数量, 理由是这些用户可能同时使用系统; 还有一种比较接近正确的观点是把在线用户数量理解为并发用户数量。实际上在线用户也不一定会和其他用户发生并发, 例如正在浏览网页的用户, 对服务器没有任何影响, 但是, 在线用户数量是计算并发用户数量的主要依据之一。

## 1.3 吞吐量 and 思考时间

吞吐量是在一次性能测试过程中网络上传输的数据量的总和。吞吐量/传输时间就是吞吐率, TPS 是每

秒钟系统能够处理的交易或者事务的数量,它是衡量系统处理能力的重要指标,是软件的性能承载能力,一般用吞吐量请求数/秒或页面数/秒来衡量。吞吐量用于协助设计性能测试场景,衡量性能测试场景是否达到预期设计目标;也用于协助分析性能瓶颈,定位性能瓶颈的位置,例如 RBI(Rapid Bottleneck Identify)方法就能通过吞吐量测试发现性能瓶颈。

作为主要性能指标,吞吐量和并发用户数之间存在关联,在没有性能瓶颈出现时,公式(3)可以计算吞吐量。其中, $F$ 是吞吐量, $Nvu$ 是虚拟用户(VU)数, $R$ 是每个VU发出的请求数量, $T$ 是性能测试使用的时间<sup>[4]</sup>。

$$F = Nvu * R / T \quad (3)$$

虽然吞吐量指标可衡量系统承受的压力,但在不同并发用户数量的情况下,对同一系统施加相同的压力,很可能得出不同的测试结果。在一简易实验中,测试A采用100个并发用户,每个VU间隔1秒发出一请求,那么,测试吞吐量 $F = Nvu * R / T = 100 * 1 / 1 = 100$ ,测试B采用1000个并发用户,每个VU间隔10秒发出一个请求,那么,测试吞吐量 $F = Nvu * R / T = 1000 * 1 / 10 = 100$ ,但在测试结果上,对于测试A,应用在50页/秒出现性能瓶颈,对于测试B,应用在25页/秒出现性能瓶。

思考时间(think time),是指用户在操作时,每个请求之间的间隔时间,特别是在交互应用中,因此,在自动化测试中,必须考虑两个操作之间的等待时间,在测试脚本中加入 think time 函数,以真实地模拟用户操作。

#### 1.4 性能计数器

性能计数器用于描述服务器的性能,包括进程时间,内存使用数,物理磁盘读写时间,成功/失败点击数,每秒点击数,尝试连接数,缓冲区命中率等。

点击率 HITPS,每秒钟用户向 WEB 服务器提交的 HTTP 请求数。这个指标是 WEB 应用特有的一个指标, HITPS 是 WEB 应用能够处理的交易的最小单位。如果把每次点击定义为一个交易,点击率和 TPS 就是一个概念,点击率越大,对服务器的压力越大,点击率只是一个性能参考指标,重要的是分析点击时产生的影响。需要注意的是,这里的点击并非指鼠标的一次单击操作,因为在一次单击操作中,客户端可能向服务器发出多个 HTTP 请求。

资源利用率,指的是对不同的系统资源的使用程度,例如服务器的 CPU 利用率,磁盘利用率等。资源利用率是分析系统性能指标进而改善性能的主要依

据。资源利用率主要针对 WEB 服务器、操作系统、数据库服务器、网络等。

## 2 性能测试实验设计

“学习交流系统”是一个典型的三层 B/S 架构系统。测试应用 MI 公司的产品 LoadRunner,进行应用服务器的压力测试,找出应用服务器能够支持的最大客户端数。测试中,逐步增加并发用户数,考察应用服务器的运行情况,分析性能指标的变化。

### 2.1 使用 LoadRunner 对 Web 应用进行负载测试

MI 公司的 LoadRunner 是性能测试的工业级标准负载测试工具,通过模拟上千万用户实施并发负载,通过实时性能测试来确认和查找问题,缩短测试时间,优化性能和加速应用系统的发布。

Web 应用负载测试包括负载测试计划制定、测试脚本开发、运行场景创建、运行测试、监视场景、分析测试结果等步骤。开发测试脚本要使用 VuGen 组件,脚本要完成的内容包括每一个虚拟用户的活动、定义结合点、定义事务。运行场景描述测试活动中发生的各种事件,利用 Controller 创建运行场景、运行虚拟用户活动的机器列表、测试脚本的列表、大量的虚拟用户和虚拟用户组,Analysis 则用于在测试结果的分析<sup>[5,6]</sup>。LoadRunner 执行性能测试可分为四个步骤:

Virtual User Generator 创建脚本:创建脚本,选择协议;录制脚本;编辑脚本;检查修改脚本是否有误。

中央控制器(Controller)来调度虚拟用户:创建 Scenario,选择脚本;设置机器虚拟用户数;设置 Schedule。

运行脚本:分析 scenario。

分析测试结果

### 2.2 负载测试场景

设计三个在线测试测试场景,用户数分为三个等级:50、100和150个用户,分别对应测试场景1、测试场景2和测试场景3。场景测试步骤如下:

用户登录,选择登录用户个数(50、100、150),所有用户都同时并发操作;

用户选择一个版块,点击“发表文章”,用户输入文章标题、正文以及验证码,点击“提交”;

用户选择某篇文章,点击“回复”输入回复内容以及验证码,点击“提交”;

点击“注销”按钮,退出系统。

### 2.3 监控指标

**事务监控:**事务监视器显示方案执行期间的事务速率和响应时间。

**网络监控:**网络监视器监控网络发送的数据包,数据包返回后,监视器计算包到达请求的节点和返回所用的时间,即网络延迟时间。

**服务器资源监控:**通过在方案运行期间监视资源,确定在特定的计算机上出现瓶颈的原因。

**web 资源监控:**使用 Web 资源监视器,分析 Web 服务器的吞吐量、方案运行期间发生的每秒点击次数、每秒 HTTP 响应数、从 Web 服务器返回的 HTTP 状态代码、每秒下载页面数、每秒服务器重试次数、打开的 TCP/IP 连接数、新的 TCP/IP 连接数,以及每秒 SSL 连接数。

## 3 实验结果分析

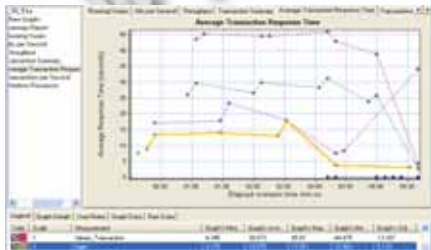


图 2 50 个并发用户的事务平均响应时间



图 3 50 个并发用户的 Windows 资源图

在测试场景的执行过程中,选择录制并保存脚本,选择加压、缓慢加压、默认计划。其中,默认计划:同时加载所有 vuser,直到完成;加压:每 15 秒启动 2 个 vuser 持续时间 5 分钟;缓慢加压:每 2 分钟启动 2 个 vuser 持续时间 10 分钟。场景运行完毕,ANALYSIS 打开结果文件(.LRR),对相关的性能指标进行分析。标准偏差值 STD 用于衡量数据的偏离程度,可以从 STD 指标看出数据是分散还是集中的,STD 越小,

数据越集中。图 2 是 50 个并发用户的事务平均响应时间图,图 3 是 50 个并发用户的 Windows 资源图。

从图中可以观察到“登录”事务的平均事务响应时间,最大值是 17.971 秒,最小值是 3.175 秒,平均值 10.575 秒,标准偏差值 5.132,数值相对比较分散,比对事务平均响应时间的标准,系统的响应还没达到预想的要求。

在测试场景 2 中,设置 100 个并发用户,事务“登录”的事务平均响应时间最大值为 30.571 秒,最小值为 3.188 秒,平均值为 15.996 秒,标准偏差值为 9.012 秒,比 50 个用户时的数据还要分散。与 50 个用户的数据比较,100 个并发用户的 Windows 资源使用并没有升高,硬盘的 Disk Time 也仅为 1.966 秒,CPU 使用率平均值为 39.734%。

在测试场景 3 中,设置 150 个并发用户,事务的平均响应时间反而下降,平均值为 12.427 秒,说明系统的稳定性存在问题,虽然响应时间有所下降,但没有达到标准。因此,系统的响应时间还有待改进。从 Windows 资源数据可以得出结论,CPU 的使用率没有超出 40%,其他的数值的变化也很平缓,系统的运行对服务器的性能没有造成瓶颈。在系统运行中,服务器的性能是很稳定的,CPU 处理时间平均值仅为 39.777%,其他的几项性能数值也较低。

负载测试是 Web 应用系统测试的核心。通过在线系统的性能测试实验,分析了系统的性能瓶颈。课题的研究也发现一些问题需要进一步研究,例如瓶颈的精确定位,性能分析的细化,测试管理的优化等。

### 参考文献

- 1 单锦辉,姜瑛,孙萍.软件测试研究进展.北京大学学报(自然科学版),2005,41(13):134 - 143.
- 2 邓小鹏,等. Web 应用测试技术进展.计算机研究与发展,2007,44(8):1273 - 1283.
- 3 杨萍,李杰.利用 LoadRunner 实现 Web 负载测试的自动化.计算机技术与发展,2007,17(1),242 - 244.
- 4 段念.软件性能测试过程详解与案例剖析.北京:清华大学出版社.2006.
- 5 Menasce DA. Load Testing of Web Site. IEEE Internet Computing, 2002,6(4):70 - 74.
- 6 Mercury Interactive Corporation. LoadRunner 使用指南.