

电子政务系统性能测试应用研究

文俊浩^{1,2} 桑春艳¹ 何盼²

(重庆大学计算机学院¹ 软件学院² 重庆 400030)

摘要 首先从电子政务系统的现状出发,分析了现有电子政务系统中存在的性能问题,并对软件测试及软件测试工具 Loadrunner 进行了相关分析研究。通过性能测试并对多次实验结果进行分析比对,找出电子政务系统中存在的系统性能瓶颈,并进行了相关优化,提高了电子政务系统的工作效率。

关键词 测试,性能,自动化系统,Loadrunner

中图分类号 TP311.5 **文献标识码** A

Research and Application of Performance Testing of E-government System Based on Struts Architecture

WEN Jun-hao^{1,2} SANG Chun-yan¹ HE Pan²

(College of Computer Science¹, College of Software and Engineering², Chongqing University, Chongqing 400030, China)

Abstract On top of investigation on current office automation (OA) systems, this paper first analyzed the performance problems of these systems. Then, it researched into fields related to software testing, including Loadrunner, an automatic tool for software performance testing. Based on the analysis and comparison of several test results from performance testing, we found the bottleneck problem which affects performance of the OA system, made optimizations to the system based on the findings and finally improved the working efficiency of OA systems.

Keywords Testing, Performance, OA, Loadrunner

Web 服务对传统的测试方法理论和技术都提出了新的挑战^[3],例如,对于相对独立的各个环节,如何规范和管理测试过程如缺陷跟踪过程等,Web 应用测试贯穿于设计、开发、运行及维护的整个生存周期^[2],本文就此展开讨论。

1 引言

软件的性能测试在软件质量保证中起着非常重要的作用。通常,软件发布报告中最重要的问题往往不是系统崩溃,也不是系统功能不全,而是系统性能下降或者处理要求达不到生产能力的要求。值得注意的是,尽管软件功能方面的测试越来越广泛,但对性能测试的重视度还远远不够。

传统的性能测试工作采用手工的方式,不能模拟大批量用户并发访问系统时的真实情况,测试人员通过手工模拟系统用户访问业务系统,效果不是很理想。随着技术手段的不断进步,目前还没有一个很完善的方法对任何的 JSP/JDBC 进行设置和调整,性能测试工作的开展主要是借助于相关的软件测试工具,或者开发相关的测试软件,进而模拟大批量用户并发使用系统的场景,从而对系统性能进行更详实的考察。

随着电子政务系统应用规模的不断扩大,政府门户系统的信息访问量也越来越大,而系统资源本身(包括硬件资源和软件资源)是有限的。面对众多的用户的访问需求,提高电子政务服务的性能成为越来越多开发人员需要考虑的重点。从体系结构上看,电子政务系统是一个多层架构,这种体系结构

在逻辑上可以划分为:表示层、业务逻辑层、数据层,并且这些层次处于不同系统平台之上。从实现技术上看,电子政务系统使用了 JavaScript, database, JSP, 导致系统实现复杂。从组成成分上看,Web 应用程序由 HTML 文档, XML 文档, 图片, JavaScript, Form, JSP, JavaBeans 等构成。从运行机制看,具有分布式、并发、动态、实时交互的特点。从运行过程看,具有不确定性特征。由于 HTTP 协议的无状态属性,每一个送到服务器端的用户请求是独立的,相互之间没有联系,使得 Web 应用的运行过程取决于用户意图及行为,由于其用户数量巨大,用户意图及行为的非一致性直接导致其运行过程的不一致。从运行环境看,Web 应用性能与环境及负载有关,如客户端浏览器缓存的位置、网络带宽、服务器端配置如内存容量大小、CPU 个数、I/O 读写速度及负载(负载类型与数量)等有关。从设计开发的过程看,具有需求不明确,设计开发时间短、变化快的特点。

本文主要以贵州省某区县电子政务系统为例,介绍 LoadRunner 在电子政务系统中的性能测试方法和过程,利用 LoadRunner 来测试其性能,对比系统在优化前后性能的差异,以检验结果是否能达到设计预期。

2 性能测试工具 LoadRunner

目前主流的性能测试工具的测试流程如图 1 所示。首先制定测试计划,然后根据测试目标录制相应的测试脚本,然后

到稿日期:2008-05-05 本文受国家支撑计划重点项目(2007BQA00316),国家自然科学基金(70672011)资助。

文俊浩(1969-),男,副教授,博士,主要研究方向为软件工程,E-mail:jhwen@cqu.edu.cn;桑春艳(1983-),女,硕士研究生,主要研究方向为软件测试;何盼(1984-),女,硕士研究生,主要研究方向为软件工程。

对脚本进行场景组织,并发回放多个测试脚本,不断地增加并发的用户数目,在进行并发性能测试的同时,对应用系统的服务器进行系统资源监控,采集并发测试过程中系统的资源使用情况,对服务器的内存、CPU、磁盘 I/O、换页操作等情况进行资源监控和分析,定位系统的瓶颈。

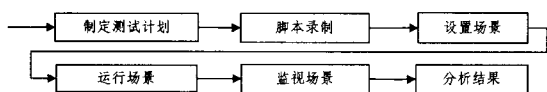


图1 测试过程

LoadRunner 是一种较高规模适应性的自动负载测试工具,它能预测系统行为,优化性能。LoadRunner 强调的是整个企业的系统,通过模拟实际用户的操作行为和实行实时性能监测,更快地确认和查找问题。

LoadRunner 包含了三个组件:

1) 提供产生压力脚本的 Vugen: 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,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。

2) 提供运行脚本并对后台进行监控的 Controller: 在 LoadRunner 的体系结构中,Player 是负载生成器,而 Controller 在整个体系结构中起到了“指挥家”的角色,即控制 Player 的运行。Player 如何运行脚本,由 Controller 来决定,Conductor 和 Player 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。

3) 对压力数据进行分析的 Analysis: 用于测试结果分析,此组件是一个数据处理组件,通过对 Controller 收集的大量数据进行综合分析,以图的形式展现给用户。用户通过分析各种参数指标,如 Web 应用的页面吞吐量、每秒点击次数和事务平均响应时间,系统资源如 CPU 利用率、内存利用率、磁盘 I/O,中间件服务器的各性能参数指标、Weblogic 的 JVM 内存使用情况、事务的响应时间、EJB 的使用情况、数据库如 Oracle 的高速缓存命中率、Sql 语句的执行情况、锁资源的急用等,来确定系统的性能瓶颈问题^[1]。

3 电子政务系统负载测试过程

3.1 测试准备

贵州省某区县的电子政务系统,其结构采用不同级别,不同单位、不同部门之后的大小 OA(Office Automation,办公自动化)流转机制,其组织结构如图 2 所示。

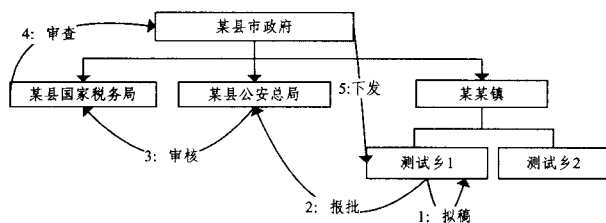


图2 公文流程

电子政务系统文件传输的流程,采用动态的自定义流程机制,有其对应的流程跟踪状态,结构比较科学、合理和高效。在图的顶端是县政府电子政务的顶层系统,大 OA 系统。乡、镇、局属于小 OA 用户层,公文、会议、档案等可以向上级单位

大 OA 递送,同时也可以在小 OA 用户之间进行流转。

在功能具备的前提下,如何保证电子政务系统能满足 24 小时不间断服务,在用户并发访问的情况下,不会出现系统崩溃、页面滞后的问题,同时保证业务支撑系统的服务质量,已经成为广大用户关心的新问题。

3.2 测试实施

3.2.1 制定测试计划

电子政务系统在大量并发用户访问下,将会成为影响系统性能的瓶颈。对于将要部署的 OA 系统,测试时使用的网络环境为局域网,通过模拟该系统的 2000 个注册用户,基本上所有的用户每天(8 小时工作时间)都会访问 OA 系统,平均在线时间(从登录 OA 系统到退出 OA 系统之间的时间间隔,也可以是多次在线时间的合计)为 12 分钟,那么该 OA 系统的平均在线数(也就是 Web 应用 Session 变量数)为 50 个($2000 * 0.2 / 8$),假设峰值在线用户数是平均在线用户数的 3 倍(该倍数可根据实际情况调整),则性能测试需求的在线用户数为 150。

由于时间和资源限制,不可能对 Web 应用系统的所有功能进行性能测试,而是从业务的角度(如某一功能操作的用户多)和技术的角度(如某一功能虽然访问用户不多,但内部处理逻辑复杂或处理数据量大)来选择 Web 应用系统的特定功能作为性能测试用例。由于所有用户都经常使用公文查询功能,因此确定的性能测试用例为公文查询。

响应时间的快慢直接影响了系统使用用户的满意度,采用平均响应时间来描述系统性能测试需求是不科学的,因为无法直接和客户的满意度挂钩。在做性能测试时,如果某一请求的响应时间过长或过短,将导致平均响应时间和实际情况偏离。以 OA 系统为例,定义的响应时间需求为:90%(该百分比和要求的系统用户满意度相关)的查询请求响应时间不超过 8 秒(该时间可根据实际情况确定)。单位时间内 Web 应用系统需处理多少笔特定的交易可通过统计获得。以 OA 系统为例,假设每个用户每天(一天按 8 小时计算)平均会查询公文 4 次,那 OA 应用的 Web 服务器平均每分钟要能处理 $2000 * 4 / (60 * 8) = 16.67$ 笔查询公文交易,考虑到峰值因素,要求每分钟能处理 $16.7 * 3 = 50$ 笔查询公文交易。

3.2.2 录制脚本

通过对测试计划进行详细的分析,针对系统的公文查询功能点,录制新建公文,批阅公文,查看公文功能的脚本。在录制的过程中,把功能点的操作设置为一个事务,方便在结果分析的过程中查看事务的运行情况,保证用户对同一个功能点的真正并发,需要设置集合点。

3.2.3 设置、运行和监控场景

在测试过程设置两种类型的场景,场景一:最大用户数为 150,每次增加 10 个用户数。场景二:最大用户数为 150,150 个用户同时运行。在开始运行场景之前,要设置监控场景中的监控量。因为系统以 Tomcat 为应用服务器,以 MySQL 为数据库服务器,所以在监控量的设置中要添加对 Tomcat, MySQL 的监控。

用户最频繁的 3 个操作,即新建公文(Xjgw_Transaction),批阅公文(Pygw_Transaction),查看公文(Ckgw_Transaction)操作。图 3 是 vuser 为 150 时电子政务系统的平

(下转第 204 页)

Web Conference Archive, Proceedings of the 13th International Conference

[7] Liang Qianhui. Composite service discovery, description and invocation. Doctoral Thesis, 2004

[8] Day J, Deters R. Selecting the best web service // IBM Centre for Advanced Studies Conference. 2004

[9] Thompson M S, Midkiff S F. Service Description for Pervasive Service Discovery // Proceedings of the First International Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI) (ICDCSW'05). 2005

[10] Bonatti P A, Festa P. On optimal service selection // Proceedings of the 14th International Conference on World Wide Web; SESSION: Service selection and metadata, 2004

[11] Yang S J H, Lan B C W, Chung Jen-Yao // Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05). 2005

[12] Antoniou G, Skylogiannis T, Bikakis A. DR-BROKERING: A semantic brokering system. Knowledge-Based Systems, Feb. 2007

[13] Simon L, Bansal A, Kona S. Towards a Universal Service Description Language // Proceedings of the International Conference on Next Generation Web Services Practices

[14] Zein O K. An Approach for Service Description and a Flexible Way to Discover Services in Distributed Systems // Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05). 2005

[15] Li Jing-shan, Liao Hua-ming, Hou Zi-feng, et al. Adynamic service composition method based on semantic interface description in pervasive computing. Journal of Computer Research and Development, 2004, 41(7): 1124-1134

[16] 李曼, 王大治, 杜小勇, 等. 基于领域本体的 Web 服务动态组合 [J]. 计算机学报, 2005, 28(4): 644-650

(上接第 180 页)
均事务响应时间图。

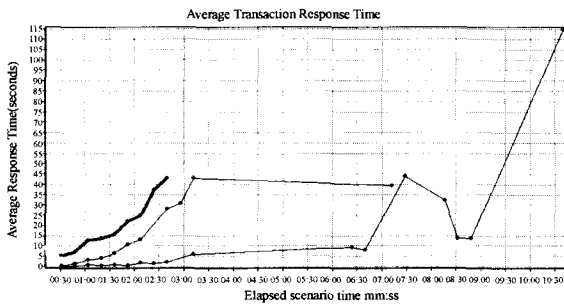


图3 Vuser 为 150 时测试的平均事务响应时间图

评价一个 Web 系统性能非常重要的指标就是操作的响应时间、CPU 的使用率等。LoadRunner 可以根据需要记录每个操作的响应时间。

4 测试结果分析

针对制定的测试性能指标,要找出在响应时间为用户可接受的范围内电子政务系统可支持的最大并发用户数。根据 Mercury 公司的报告,网站性能问题可能由许多因素引起^[4]。但是,大约一半的性能问题是由于网络、Web 应用程序、Web 服务器和数据库服务器引起的。对数据库操作依赖性很大的动态网站出现性能问题的风险尤其大。

电子政务系统在大量并发用户访问下,首先是硬件原因,包括 CPU 的主频、内存的容量以及网络带宽影响系统性能的瓶颈。如果排除硬件原因的话,分析是否是应用软件和系统软件影响了系统的性能。根据图 3 的结果,在高并发的情况下,操作的响应时间过长,按照定位的性能瓶颈,依次检查 Web 程序、数据库和应用服务器配置。

测试时使用的网络环境为局域网,硬件配置较差,而系统正式使用时的网络环境为 100Mbps 电信和网通专线接入,因此可以排除网络对系统性能的影响。数据库服务器使用的配置较高,并且测试时数据库服务器的 CPU 占用率和内存使用率均保持在正常状态,因此也可以排除数据库服务器对性能的影响。

如表 1 所列,在并发数较多的情况下,出现了较多 Web 服务器连接超时和拒绝请求的情况,判断是由于 Web 服务器

连接人数限制造成的,找到了系统性能瓶颈。某些事务响应时间仍然较高,但这是每个测试脚本模拟 150 个客户端瞬时访问数据库的情况,而且在这种情况下,应用服务器和数据库服务器均工作正常。

表 1 平均响应时间

Transaction Name	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
Xjgw Transaction	0.282	7.376	18.454	5.226	14.075	144	1	0
Pygw Transaction	4.784	13.716	27.197	5.738	24.389	145	1	0
CKgw Transaction	4.275	13.99	36.015	7.347	26.239	147	2	0

通过以上分析,优化数据库设计,检查表的索引,发现部分表的索引设计不合理,为部分表的字段增加索引。对个别频繁访问多表的过程,在主表内增加部分冗余数据,避免多表查询。进行系统调优后,对 Vuser 为 150 的情况再次进行了测试,优化后性能得到大幅提升。

结束语 负载测试结果很难仅仅解释为通过或失败,需要根据系统的主要功能来选择合理的评估指标,发现瓶颈的目的主要是为了掌握系统,为改善和扩展系统提供依据。对这些问题的充分研究及解决必将有助于 Web 应用测试技术的进一步发展。

参考文献

[1] 桑圣洪,胡飞.性能测试工具 LoadRunner 的工作机理及关键技术研究[J].科学技术与工程,2007,6(7):1019-1024

[2] 邓小鹏,邢春晓,蔡莲红.Web 应用测试技术进展[J].计算机研究与发展,2007,44(8):1273-1283

[3] 白晓颖,赵冲冲,戴桂兰.Web 服务测试研究[J].计算机科学,2006,33(2):252-256

[4] 熊忠阳,李光勇,张玉芳,等.Web 集群系统性能测试与优化[J].计算机应用研究,2008,25(3):826-832

[5] Tsai W T, Zhou Xinyu, Paul R A, et al. A Coverage Relationship Model for Test Case Selection and Ranking for Multi-version Software. [J] // 10th IEEE High Assurance Systems Engineering Symposium. 2007: 105-112

[6] Beydeda S, Gruhn V. State of the art in testing components [C] // Proc. of the 3rd Int'l Conf. on Quality Software (QSIC'03). Los Alamitos, CA: IEEE Computer Society Press, 2003: 146-153