Web 性能测试的研究与应用

江新，江国华

（南京航空航天大学　信息科学与技术学院，江苏　南京 210016）

摘 要： 性能是衡量一个 Web 应用程序好坏的重要方面,而性能测试则是保证一个 Web 应用程序成功运行的重要手段。本文针对实际参与的企业内部信息安全监控系统的测试项目，利用 OMT 模型对此系统进行建模、构建性能测试用例，并结合自动化工具 LoadRunner 对其进行性能测试。实践表明，利用 OMT 模型对此系统进行建模，有利于测试用例的提取，能够快速定位系统的性能瓶颈。

关键字：性能测试；测试用例；LoadRunner；OMT 模型

Research and Application of Web Performance Test

JIANG Xin, JIANG Guo-hua

（College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China）

Abstract: Performance is an important aspect of measuring Web Application, and performance testing is an important method of ensuring its working successfully. For an enterprise's internal information safety monitoring system's test project, the paper uses OMT model to model the system, constructs performance test cases and at last uses performance test tool LoadRunner to test the system's performance. Practice shows that using OMT model is convenient to extract test cases and locate performance bottleneck quickly.

Keywords: performance testing; test case; LoadRunner; OMT Model

1．引言

随着 Internet 的迅速发展，各种网络服务层出不穷，出现了大量基于 B/S 结构的 Web 应用系统。在这些 Web 应用中，客户端只完成一些简单的如浏览、查询、表单输入等请求操作，而绝大部分的计算处理工作则由服务端完成，这就使得 Web 服务器的负担很重。因此要求 Web 应用系统具有较高的性能。性能是衡量 Web 系统好坏的一个重要因素。而 Web 性能测试则是保证其性能的重要手段，通过性能测试能够发现影响系统性能的瓶颈，从而可以有效地指导系统的调优活动，确保将来系统投入运行之后的安全性、可靠性和执行效率。

本文以一个企业内部信息安全监控系统（以下称为 TMMS 系统）作为测试对象，使用了一种用于 Web 系统性能测试的对象模型，即 OMT（Object-based Model for Testing Web Applications）模型，并以此模型为基础设计测试用例，选用自动化测试工具 LoadRunner 模拟产生大量的虚拟用户，对系统进行性能测试。确定系统最大支持的并发用户数量，根据测试结果分析系统的瓶颈和问题所在，为系统的进一步优化提供参考。

2．Web 应用系统的性能测试

2.1 性能测试的目标

性能测试是一种信息的收集和分析过程，在这个过程中收集的数据用来预测怎样的负载水平将耗尽系统资源。主要是通过性能测试来考察在不同的用户负载下，Web 系统的响应情况和处理方式，以确保系统投入使用前的安全性、可靠性及执行效率。性能测试是在可靠性和正确性的基础上侧重效率方面的验证。通过 Web 性能测试能够发现系统的软件问题和硬件瓶颈，并提供一定量的数据来帮助分析和查明问题所在，最后达到优化系统性能的目的。

2.2 性能测试方法

针对不同的应用类型和侧重点，目前主要有以下 3 种 Web 性能测试方法：

1）虚拟用户方法：通过模拟真实用户的行为来对被测程序（application under test AUT）施加负

载，以测量 AUT 的性能指标值，如事务的响应时间、服务器的吞吐量等。它以真实用户的“商务处理”作为负载的基本组成单位，用“虚拟用户”来模拟真实用户。

2）WUS 方法：基于“网站使用签名（Web site usage signature WUS）”的概念来设计测试场景，以“经常被访问的路径”作为负载的基本单位。WUS 的提出是为了衡量测试负载和真实负载之间的接近程度，但是负载的确定依赖于日志。

3）对象驱动方法：基本思想是将 AUT 的行为分解成可测试的对象，对象可以是链接、命令按钮、消息、图像、可下载的文件、音频等。对象定义的粒度取决于应用程序的复杂性。

3．OMT 模型和 TMMS 系统概述

3.1 OMT 模型

OMT 模型，即对象模型，是在以上介绍的对象驱动方法基础上产生的。在这个模型中，Web 应用程序被视为由一系列的实体组成，如 Web page、 脚本、使用不同程序语言开发的组件等。并且每一个实体都被建模成具有属性和行为的对象。属性既可以是程序变量也可以是文档元素，而行为则指的是那些用脚本语言编写的函数或方法。

OMT 模型根据对象及对象之间的关系描述了 Web 系统的内部结构。为了限制 OMT 模型的范围及减轻此模型的复杂程度，我们只对那些对 Web 应用程序而言有意义的超链接和 Web 页面进行建模，而不考虑那些导向外部 Web 页面的超链接。为了便于在测试中捕获信息，将 Web 应用程序分为以下三类对象：

1）client 端对象：指的是在客户端浏览器中被解析并显示的 HTML 页面或 XML 文件。这类对象的属性由文档元素组成，而文档元素又可以是一个表元素、脚本或客户端对象自定义的数据对象。

2）server 端对象：指的是在 HTTP 响应中被调用的服务器端的脚本或程序，如各种的 CGI 脚本，ASP 页面，JSP 页面等。

3）components 对象：可以是下列条件之一的对象、程序或数据文件：client 端或 server 端对象中引用了的属性或方法；server 端对象调用的程序；server 端对象引用的数据文件；components 对象所引用的对象或程序。

以上的三类对象之间的关系有以下七种：继承、聚合、联合、请求、响应、导航和重定向。其中后四种关系用于描述 client 端和 server 端对象之间的关系。

整个的 OMT 模型是一个有向图，其中节点表示对象，边表示各对象之间的关系。如下图 3-1 所示就是用于 Web 应用程序的 OMT 模型。

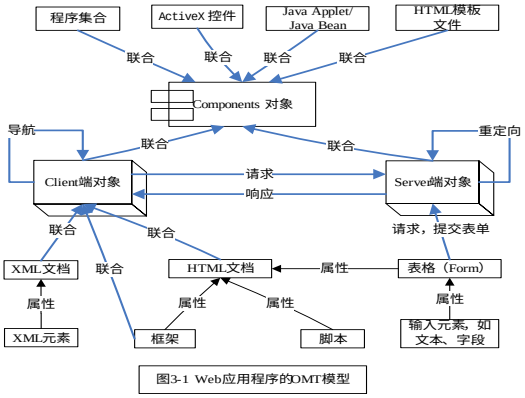

图3-1 Web应用程序的OMT模型

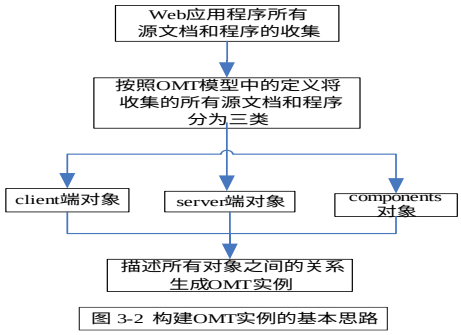

图 3-2 构建OMT实例的基本思路

3.2 OMT 模型的实例构建方法

在实际应用中，我们依据 OMT 模型，对 Web 应用系统中的各种实体，如源文档、程序、Web 页面进行分析，确定此 Web 系统的三类对象及对象间的关系，从而得出特定的 Web 应用系统的 OMT 模型实例。实例构建的基本思路如下图 3-2 所示。

3.3 TMMS 系统介绍

3.3.1 企业信息安全监控系统—TMMS 系统

本系统主要用在企业内部，由 TMMS Server 来监控和管理各个注册用户，保障企业信息安全。整个的 TMMS 系统由两大部分组成，其一是 client 端，也就是安装有 TMMS 客户端产品的智能手机，其二是 server 端，由 Web server、Application server 以及 DB server 组成。其中 Application server 是 TMMS Server 的核心，又分为 Master server(主服务器，负责处理绝大多数事务)和 MSCM(代理服务器)两部分。在企业内部用户可以通过 Wi-Fi/ ActiveSync 直接与 Master server 通信，而在企业外部，则是通过 MSCM 与 Master server 进行通信。

在针对本系统的性能测试中，我们主要关注的是 Master Server 的性能， Master Server 具有以下功能模块：

1） 设备管理模块，该模块负责管理设备的注册和更新，包含以下几个方面：Status（可以查看所有未注册或已注册设备的状态）、Task(用于 server 端向客户端发送命令，比如让客户端主动向 server 注册、更新，远程锁定客户端、远程使客户端格机等)、Domain Policies（在此可以进行各种设置，比如对客户端的一些指定扩展名的文件、PIM 信息进行加密，对客户端设置短信过滤功能、禁用客户端的某些功能如蓝牙、无线、USB 接口等）、logs(负责记录来自客户端上传的各种 log)、Manage Device(负责设备的增、删、改等操作)。

2）Updates 模块负责 server 端、客户端组件的更新。

3）Administration 模块，负责所有 log 的维护、SMS Sender 的配置等。

3.3.2 OMT 模型用于 TMMS 系统性能测试的优势

在对 Web 系统进行性能测试时，通常需要测试系统在大量并发用户同时访问下的性能情况，由于受客户端条件限制，不可能同时使用几千台真实的装有 TMMS Client 的智能手机来做测试，故通过编写一个内嵌有表格、交互式表单、CGI 请求脚本等的 html 文件来模拟 client 端的主要行为。

当 client 向 server 发送 http 请求（如注册、更新）时，这些 http 请求中包含处理请求的 CGI 扩展程序的 URL 值。当 server 收到这些请求后，就会依据 URL 值调用相应的 CGI 程序来处理，而 client 发送的是以 html 页面为载体的表格、client 端自定义的数据对象和内嵌的脚本等。TMMS 系统中的 HTML 请求页面、CGI 程序、各种程序模块均可以用 OMT 模型中的对象来表示，这样对系统进行建模以后，系统的组织结构就非常的简明，当对系统进行性能测试时，就能较容易的提取测试用例。

4．基于 OMT 模型的 TMMS server 端性能测试

4.1 TMMS 系统中的三类对象

根据 OMT 模型中三类对象的定义，TMMS 系统中的三类对象描述如下：

Client 端对象：以 html 页面为载体发送至 server 端的各种 http 请求页面以及 server 端 http 响应的 html 页面；Server 端对象：在 http 响应中被调用的各个 CGI 程序；Components 对象：server 端对象中的各个 cgi 程序、client 更新时 server 端调用的 AU server 中的 server.ini 文件、以及与 server 有关的各种配置文件（Config files）、系统管理员操作 server 端时用到的 ActiveX 组件等。

下面给出了使用 OMT 模型构建的 TMMS 系统的部分模型，此部分实现的是用户的注册和更新操作，如图 4-1 所示：

由图 4-1 我们可以很清楚的看到，client 端对象、server 端对象、components 对象之间的关系及通信方式。这样在对 server 进行性能测试时，就能以 server 端的对象为目标设计测试用例。以 client 向 Master server 注册为例，我们可以通过上图中的 Register 页面这一客户端对象来测试 server 端与处理用户注册有关 CGI 程序集合。Register 页面这一对象具有三种方法，功能依次是向 server 端发送与注册有关的 register、getSetting、reportVersion 三个 CGI 请求。这些 CGI 请求到达 Master server 后，将由 Master server 调用相应的 CGI 程序来处理,并给 client 端一定的反馈信息。这样通过测试 server 端用于处理用户主业务流程的对象集合的性能，就可在一定程度上反映 server 端的性能状况，且能很容易的定位系统的性能瓶颈。

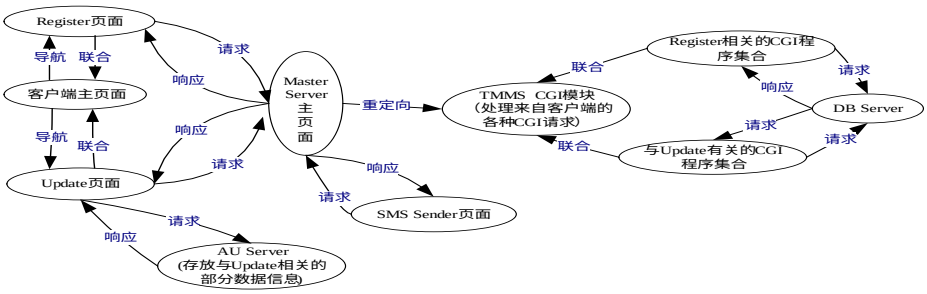

图4-1 注册、更新模块的DMT表示

4.2 TMMS server 端的性能测试

4.2.1 测试计划的制定

对此系统性能测试的总体目标是验证和确认此系统的性能（用户注册、更新的成功率、最大并发用户数等）能满足 1500 用户规模的使用需求，并判断能否满足更多用户规模的需求。在进行性能测试时，选择的测试工具是 LoadRunner，利用它捕获 client 和 server 之间的通信信息，并转换成 LR 脚本，对脚本进行增强以后，模拟成百上千的用户对 Master server 实施并发负载测试，同时监测系统的性能。

4.2.2 设计测试脚本

下面以用户注册为例说明测试脚本的设计.在 4.1 节中，我们提到了用户注册由三步组成，利用 LoadRunner 可以录制这三步的操作流程，并转换为三个相应的 LR 脚本。对注册而言需要编写以下脚本来控制已录制的三个脚本的执行顺序，具体脚本设计如下：

```
WholeFlow_register ()
{    int action_result=LR_FAIL;
 lr_start_transaction("Action_register ");
 Action_register (); action_result=lr_get_transaction_status("Action_register ");
 lr_end_transaction("Action_register ", LR_AUTO);
 if(LR_FAIL==action_result) goto END_WITH_FAIL;
 lr_start_transaction("Action_getSetting");
 Action_getSetting (); action_result=lr_get_transaction_status("Action_getSetting;
 lr_end_transaction("Action_getSetting", LR_AUTO);
 if(LR_FAIL==action_result) goto END_WITH_FAIL;
 lr_start_transaction("Action_reportVersion");
 Action_reportVersion (); lr_end_transaction("Action_reportVersion", LR_AUTO)
END_WITH_FAIL;      return 0;}
```

上面三步中的每一步都对应一个独立的脚本，这三个脚本可以通过 LR 录制 client 与 server 端之间的通信过程并加以脚本的增强得到。脚本内容较多，这里就不再赘述了。

4.2.3 测试用例的编写

同样以用户注册为例，可以通过 Register 页面来测试 server 端处理用户注册请求的 CGI 程序模块的处理能力，这些程序既可以视为 server 端对象，也可以视为 components 对象。为了考察系统处理用户注册业务的能力，可编写类似如下的测试用例：

Title: 100 Vusers register to Master server

Steps: 1. using LoadRunner record a vuser’s whole registration process (including register、getSetting and reportVersion process)to Master server and convert it to script;2. using tool and script simulate 100 vusers registration to Master server simultaneously;3. monitor server’s performance and response 4.check log file.

Results: Server can handle it correctly and all the vusers can register successfully.

4.2.4 执行测试并分析测试结果

按照编写的测试用例，使用 LoadRunner 加载测试脚本、设置测试场景，执行测试用例，对 server 端进行性能测试。表 1、表 2 是关于 Register、Update 业务流程在不同并发用户数下测试情况数据分析的结果：

表 1：

不同数目并发用户向 Master Server 注册的结果								
并发用户数	1000	2000	3000	4000	4500	5000	5500	6000
平均成功数	1000	2000	3000	4000	4500	5000	5496.5	5945
成功率	100%	100%	100%	100%	100%	100%	99.94%	99.08%

表 2：

不同数目并发用户向 Master Server 更新的结果								
并发用户数	500	1000	1500	2000	3000	4000	5000	6000
平均成功数	500	1000	1355	1621	2745	2835	3846	3822
成功率	100%	100%	90.30%	81.10%	91.80%	7.0.9%	77.30%	63.70%

由上面的结果可以看到：当并发用户数不超过 5000 时，用户基本上都能注册成功，但是当并发用户数超过 5000 时，会有部分用户注册不成功，且随着并发用户数的增加，注册不成功的用户数也会增加，成功率降低；对于 Update（此过程类似于 Register）而言，当并发用户数不超过 1000 时，成功率是 100%，但当并发用户数超过 1000 后，成功率开始下降。经分析日志文件及相关数据得出 Update 的瓶颈在向 AU Server 发送请求获取 server.ini 文件这一步。定位性能瓶颈以后，采取一些策略，对 server 端进行调优，从而提高系统性能。

5．结束语

针对 Web 系统的性能测试是一项复杂的工作，本文使用 OMT 模型，对 Web 应用系统进行建模，不仅有助于测试人员对 Web 系统结构的理解，而且有助于测试用例的提取。在实际应用中，将 OMT 模型引入企业信息安全监控系统（TMMS 系统）的性能测试中，同时配合使用性能测试工具 LoadRunner 对该系统进行性能测试。此方法在实际的应用中，避免了许多盲目行为，有效地提高和改善了 Web 应用性能测试的效率。

参考文献

[1]. MI Corporation. Load testing to predict Web performance[M] .Mountain View, USA: Mercury Interactive Corporation, 2003.

[2]. Subraya B M, Subrahmanya S V. Object Driven Performance Testing of Web Applications [C]. Quality Software. Proceedings of the 1st Asia-Pacific Conference, 2000. 17-26.

[3]. 谭浩.性能测试的原理及其自动化工具的实现[J].计算机工程与设计, 2006, 27(19): 3660 -3662.

[4]. 赫建营. 一种有效的Web性能测试方法及其应用[J].计算机应用研究, 2007,22(1): 275 -277.

[5]. 马琳,罗铁坚,宋进亮.Web性能测试与预测.中国科学院研究生学报,2004:472-479.

[6]. 许蕾，徐宝文.Web应用测试框架研究[J].福建：东南大学学报，2004:8-9.

作者简介

江新，女,硕士，主要研究方向：软件工程，软件测试

江国华，男，副教授，主要研究方向：软件工程，软件测试

Web性能测试的研究与应用

作者： 江新， 江国华
作者单位： 南京航空航天大学信息科学与技术学院,江苏 南京 210016