**实用软件测试性能测试教学案例使用说明**

**——以CRM系统压力测试为例**

1. **教学目的与用途**

本案例主要适用于软件工程专业，也适用于计算机技术专业的软件测试技术学习。本案例的教学目的是通过对CRM系统实施压力测试，使学生了解如何使用压力测试工具PR，并且通过该案例熟悉压力测试过程中的安装、脚本录制回放、关联、参数化、事物、集合点、IP欺骗、场景设置、报告分析等技能。

**2． 涉及知识点**

压力测试、脚本录制回放、关联、参数化、事物、集合点、IP欺骗、场景设置、报告分析

1. **配套教材**

Implementing Automated Software Testing [J]. 机械工业出版社，北京：2010.04

1. **启发思考题**
2. 结合web形式的系统应用CRM，可以使用什么自动化测试工具？它们各自有何缺点？
3. PR的主要功能和使用流程？
4. PR的主要元件？
5. 思考怎样使用事物和时间点？怎么设置场景才能更好地执行压力测试，得到更有价值的数据？
6. 尝试自学JMeter，并与PR全面比较。
7. **分析思路**

本案例给出一个完整训练学生掌握压力测试的被测软件和测试脚本、相关文档。CRM是本案例的AUT（Application Under Test），是基于web的客户关系管理系统。Web性能测试就是模拟大量用户操作给网站造成压力，并评测Web系统在不同负载和不同配置下能否达到已经定义的标准，分析和消除与软件结构中相关联的性能瓶颈。步骤如下:安装PR（Spasvo Performance Runner）,建立测试环境；设置和更改工作空间；新建项目；录制脚本和回放脚本；脚本编程，包括关联、事物、集合点、IP地址欺骗、脚本参数化等；新建场景，设置场景；报告分析。

1. **理论依据及分析**

Web应用性能测试是为了描述Web应用系统与性能相关的特性并对其进行评价，而实施和执行的一类测试。Web应用性能测试主要检验软件是否达到需求规格说明中规定的各类性能指标，并满足一些性能相关的约束和限制条件。简而言之，Web性能测试就是模拟大量用户操作给网站造成压力，并评测Web系统在不同负载和不同配置下能否达到已经定义的标准，分析和消除与软件结构中相关联的性能瓶颈。

Web应用性能测试包括负载测试和压力测试两个方面，在进行时都需要采用相应的工具。目前比较流行的负载测试和压力测试工具有LoadRunner、WebLoad、QALoad、Apache JMeter等。响应国家的软件国产化技术要求，本案例使用上海泽众公司(SPASVO)的性能测试工具PR（Performance Runner）进行压力测试。

PR是一款自动化压力测试工具，通过加载不同的测试组件，实现面向不同协议的测试，通过模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标，找出性能方面的问题，发现性能瓶颈，优化设计。

PR自动测试工具适用于常规压力测试、极限压力测试、负载测试、可靠性测试等，可以提高测试效率，降低测试人工成本，帮助用户找被测对象的缺陷，特别是对于一些通过手工测试很难发现的缺陷。

PR可以进行基于HTTP协议的性能测试，一般为B/S架构的Web程序基于SOKCET协议的性能测试，一般为C/S架构的桌面程序。

本案例的AUT是基于web的客户关系管理系统，采用PR进行压力测试可以满足需求。

**6.1 Web性能测试工具Apache JMeter**

Apache JMeter是一个专门为进行服务器装载测试而设计的、100％的Java桌面应用程序，用于对C/S结构的软件(例如web应用程序)进行负载测试和压力测试。

Apache JMeter是100％的Java桌面应用程序，它被设计用来加载被测试软件功能特性、度量被测试软件的性能。设计Jmeter的初衷是测试Web应用，后来又扩充了其它的功能。Jmeter可以完成针对静态资源和动态资源（讹误女监, Servlets, Perl脚本, Java对象, 数据查询s, FTP服务等）的性能测试。。 Jmeter可以模拟大量的服务器负载、网络负载、软件对象负载，通过不同的加载类型全面测试软件的性能。Jmeter提供图形化的性能分析。使用环境: Solaris, Linux, Windows (98, NT, 2000)，JDK1.4以上。

1.支持对应用程序进行功能/回归测试。

2.允许使用正则表达式创建断言。

3.在设计阶段，JMeter能够充当HTTP Proxy(代理)来记录IE/Netscape的HTTP请求，也可以记录Apache等Web Server的log文件来重现HTTP流量。

4. JMeter还提供可视化组件和报表工具，用于将服务器在不同压力下的 性能展现出来。

**6.2 Load Runner**

LoadRunner是工业标准级负载测试工具，是一种预测系统行为和性能的负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner 能够对整个企业架构进行测试。通过使用LoadRunner ，企业能缩短测试时间，优化性能和加速应用系统的发布周期。

LoadRunner 是一种适用于各种体系架构的自动负载测试工具，它能预测系统行为并优化系统性能。LoadRunner 的测试对象是整个企业的系统，它通过模拟实际用户的操作行为和实行实时性能监测，来帮助您更快的查找和发现问题。此外，LoadRunner 能支持广泛的协议和技术，为您的特殊环境提供特殊的解决方案。

您可以利用它的日程计划服务来定义用户在什么时候访问系统以产生负载。这样，您就能将测试过程自动化。同样您还可以用Controller 来限定您的负载方案，在这个方案中所有的用户同时执行一个动作---如登陆到一个库存应用程序----来模拟峰值负载的情况。另外，您还能监测系统架构中各个组件的性能---- 包括服务器，数据库，网络设备等----来帮助客户决定系统的配置。

LoadRunner 通过它的AutoLoad 技术，为用户提供更多的测试灵活性。使用AutoLoad ，可以根据目前的用户人数事先设定测试目标，优化测试流程。

LoadRunner还有定位性能问题，LoadRunner 内含集成的实时监测器，在负载测试过程的任何时候，可以观察到应用系统的运行性能。

利用LoadRunner 的ContentCheck TM ，用户可以判断负载下的应用程序功能正常与否。

一旦测试完毕后，LoadRunner 收集汇总所有的测试数据，并为用户提供高级的分析和报告工具，以便迅速查找到性能问题并追溯原由。

**6.3 WebLoad Analyzer**

Radview公司的TestView系列Web性能测试工具和WebLoad Analyzer性能分析工具，旨在测 试Web应用和Web服务的功能、性能、程序漏洞、兼容性、稳定性和抗攻击性，并且能够在测试 的同时分析问题原因和定位故障点。

整套Web性能测试和分析工具包含两个相对独立的子系统：Web性能测试子系统Web性能分析子系统。其中Web性能测试子系统包含3个模块：TestView Manager、WebFT以及WebLoad。 Web性能分析子系统只有WebLoad Analyzer。

WebLoad模拟多个用户行为进行测试，所测试的是系统性能，容量，稳定性和抗攻击性；WebFT 模仿单一用户行为进行测试，所测试的是系统功能，漏洞，兼容性和稳定性； WebLoad Analyzer对Web服务、中间件和数据库进行监控和分析，找出问题原因和故障点。

**6.4 Spasvo PR**

上海泽众软件科技有限公司是专业从事软件测试、软件生命周期管理技术的研究与开发、服务的高新技术企业，是国内专业测试工具软件厂商，在金融、电信、政府、研究所、大学、企业等行业拥有几千家用户。

PerformanceRunner(简称PR）是国内专业商业化性能测试工具，通过模拟海量用户并发测试整个系统的承受能力，实现压力测试、性能测试、配置测试、峰值测试等。大限度地缩短测试时间，优化性能和加速应用系统的发布周期。

PR分为三个模块：生成器，执行器，分析器。

生成器录制脚本，通过监听应用程序的协议和端口，录制应用程序的协议和报文，创建测试脚本； 编辑脚本，可以在脚本中添加校验点、集合点并实现参数化，可以大程度上满足测试需求；回放脚本，PR采用java作为标准测试脚本，对已经录制的脚本在PR中再次运行，通过回放，可以排除录制脚本时可能产生的错误，为下一步执行场景做好铺垫。

执行器设计并运行场景，场景设计视图中包含了三个部分，分别是场景组管理、预期指标管理、场景计划管理，可以为该场景添加虚拟用户的数量以及虚拟用户启动、停止的规则，还可以设置IP欺骗，远程监控服务器等。设计场景完成后，即可运行场景，场景运行界面分为五个部分，主要包括用户状态区、事务统计区、图表树、性能波形图、性能数据统计分析。可以看到实时的数据统计图如，VU图、事务图、WEB资源图、被测系统性能监视图等。

分析图可以帮助您确定系统性能并提供有关事务及Vuser的信息。通过合并多个负载测试场景的结果或将多个图合并为一个图，可以比较多个图。分析器主要通过图表：虚拟用户图、事务概要图、事务响应时间图、每秒事务图、事务性能概要图、 每秒点击量等进行常用性能分析。

1. **关键要点**

本案例的关键点有几个，即：脚本录制回放、关联、参数化、事物、集合点、IP欺骗、场景设置、报告分析。

训练难点和重点在于报告分析，即测试结果分析与性能瓶颈的识别，应得到最佳并发用户数和最大并发用户数。

案例从以下几个部分进行：

* 理解“性能”和“性能测试”
* 性能测试的过程与最佳实践
	+ 获取有效的性能需求
	+ 构建性能测试环境
	+ 设计和实现测试场景
	+ 执行性能测试
	+ 性能测试结果分析与性能瓶颈的识别
	+ 改善系统性能
* 相关主题讨论
1. **建议课堂计划**

课前计划：提出启发思考题，请同学在课前完成软件安装和初步思考。

课中计划：简要的课堂前言，明确主题 (5-8mins)

 分组讨论 (30mins)，告知发言要求

小组发言 (每组5-8mins)

引导全班进一步讨论，并进行归纳总结 (20-30mins)

课后计划：请同学在网络上找出合适的AUT（application under test），使用Spasvo PR进行压力测试，并给出结论。