**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**

**摘 要**

压力测试（Stress Test）是模拟实际应用的软硬件环境及用户使用过程的系统负荷，长时间或超大负荷地运行测试软件，来测试被测系统的性能、可靠性、稳定性等。本案例以基于web的CRM系统为AUT（application under test）,使用国产软件Spasvo Performance Runner进行性能测试中的压力测试，训练学生识别软件性能瓶颈，改善系统性能，掌握压力测试技能，学生可以从案例中学习到压力测试的各种技能，包括脚本录制回放、参数化、场景设置和报告分析。

**关键词：**压力测试；脚本录制回放；参数化；场景设置；报告分析

**Abstract**

Stress test is to simulate the software and hardware environment of practical application and the system load of user's using process, run the test software for a long time or a large load to test the performance, reliability, stability, etc. of the tested system. In this case, the CRM system based on web is aut (application under test), Use the domestic software spasvo performance runner to conduct stress test, train students to identify software performance bottlenecks, improve system performance, and master stress test skills. Students can learn various skills of stress test from cases, including script recording and playback, parameterization, scene setting and report analysis.

**Key Words:** Stress test; script recording and playback; parameterization; scene setting; report analysis

**0 引 言**

软件开发和使用的历史已经留给了我们很多由于软件缺陷而导致的巨大财力、物力损失的经验教训。这些经验教训迫使软件测试工程师必须采取强有力的检测措施来检测未发现的隐藏的软件缺陷。

软件测试大致分为软件测试管理、软件测试技术方法、软件测试自动化三类，每类可细化为更多子方面，例如第二类根据测试类型还可细化为功能测试、性能测试、安全测试等，根据测试方法可细化为黑盒测试、白盒测试、灰盒测试等。

软件测试自动化既包括技术方法方面，又包含管理方面；更重要的是，软件测试自动化是软件测试领域无法逾越的发展阶段，随着应用软件程序规模的不断扩大，业务逻辑的不断复杂，以及从业者相互协作关系的日益重要，在软件的测试活动里适当使用自动化测试是非常必要的。

执行性能测试，使用手工方式是无法想象的，因此借助工具来实现是非常必要的。目前业内存在两种现状：一是很多公司为了节约购买工具的成本或本身不要求软件性能指标而干脆不执行性能测试；二是由于性能测试是一门博大精深的技术工作，起步较高，因此这方面的高手不多，造成很多大中型软件企业或外企严重缺乏性能测试工程师！

软件开发技术越来越高深，业务逻辑越来越复杂，对软件的质量要求同样也会越来越高，软件一定会存在性能缺陷，因此对软件的性能要求也会随之而来；况且，软件的性能指标是软件用户手册里的重要组成部分，从正规测试流程上来说，凡是网络应用软件，不可不做性能测试！但是，从事性能测试的工程师，需要掌握太多的知识，包括计算机网络、数据库、操作系统、服务器等，而且还要有深厚的性能测试计划、设计、分析能力，以及丰富的性能测试经验，这些如果单靠个人的自行摸索，肯定是不太实际的。

作为软件工程必备的技能，本案例使用国产软件Spasvo PR进行性能测试中的压力测试，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。

**1 相关技术介绍**

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。

**1.1 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。

**1.2 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。

**1.3 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。

**1.4 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。

**2 对CRM进行压力测试**

**2.1实验环境**

使用SPASVO上海泽众的PR测试基于web的CRM，实验环境如下:

（1）硬件环境：

a PC机一台

b 内存要求：不少于2G

c 磁盘空间要求 ：不少于1G剩余磁盘空间

（2）软件环境：

a 操作系统要求：Windows(32位/64位) xp/7/server

b IE浏览器要求：IE6以上

c 编译环境：JDK1.6以上版本 （建议使用JDK1.6以上版本）

**2.2 软件安装和测试环境设置**

（1）检验JDK是否已安装在PC机上

命令提示行输入“java -version（查看JDK版本信息）”，按回车若提示“‘java ’不是内部或外部命令，也不是可运行的程序或处理文件”，则说明JDK未安装；若出现相应的界面，可以查看到JDK的版本信息，则说明JDK已经安装，无需再次安装。

（2）JDK安装

选择安装目录安装过程中会出现两次安装提示。第一次是安装 jdk ，第二次是安装 jre 。建议两个都安装在同一个java文件夹中的不同文件夹中。（不能都安装在java文件夹的根目录下，jdk和jre安装在同一文件夹会出错）

（3）PR的安装及初次使用

 数据库选择使用MySQL，设置安装路径，安装完成后设置PR的工作空间路径，点击文件—新建—项目，在项目下自动生成3个空的脚本文件。

* Action.bsh：存放实际的操作脚本，可多次执行，可创建多个Action脚本
* Init.bsh:一般用于存放应用程序初始化脚本
* Uninit.bsh：一般用于存放应用程序注销或关闭脚本

**2.3 脚本录制回放**

PerformanceRunner特点

● 使用BeanShell语言作为脚本语言，使脚本更少，更易于理解。BeanShell语法自身也兼容Java语法。

● 采用关键字提醒、关键字高亮的技术，提高脚本编写的效率。

● 提供了强大的脚本编辑功能。

● 具有优秀的录制功能，能够一次录制非常完善的脚本和资源，降低了测试人员修改脚本的工作量。对于不熟悉编程的测试人员来说，是非常有价值的。

● 支持各种需求的校验。包括对如header字段的各项属性，服务器返回的内容、数据库、Excel表格、正则表达式等的校验。

● 支持参数化，同时支持数据驱动的参数化。

● 支持测试过程的错误提示功能。

● 丰富的命令函数，有利于测试人员进行各种功能测试，熟练掌握这些命令函数，能够让测试人员编写出更简练、更高效的测试脚本。

步骤分别是工作空间设置、新建项目、录制脚本、回放脚本。

在录制好脚本后，在工作空间的项目目录下会存在三个bsh脚本文件及三个对应的xls参数化文件。脚本文件用来保存脚本，其中自动录制的脚本会放入Action.bsh中另外两个脚本文件中放置的是初始化代码和结束时的各种资源清理代码，这些代码由用户在需要时手工添加，默认时文件中没有脚本；参数化文件用来保存参数化数据，默认时文件中没有数据，上面的文件都可以在软件中修改，不建议在软件外编辑。

Record文件夹中保存了录制时的请求响应数据，data文件中包含了请求和响应的header数据，requ文件中包含了请求的body数据，resp文件中包含了响应的body数据，repository文件中包含了文件序号和url链接的对应关系。

**2.4关联**

关联（correlation）：在脚本回放过程中，客户端发出请求，通过关联函数所定义的左右边界值（也就是关联规则），在服务器所响应的内容中查找，得到相应的值，以变量的形式替换录制时的静态值，从而向服务器发出正确的请求，这种动态获得服务器响应内容的方法被称作关联。其实关联也属于一同特殊的参数化，只是与一般的参数化有些不同一般的参数化的参数来源于一个文件、一个定义的table、通过sql写的一个结果集等，但关联所获得的参数是服务器响应请求所返回的一个符合条件的、动态的值。

当客户端的某个请求是随着服务器端的响应而动态变化的时候，我们就需要用到关联。当然我们在录制脚本时应该对测试的项目进行适当的了解，知道哪些请求需要用到服务器响应的动态值，如果我们不明确哪些值需要做关联的话，我们也可以将脚本录制两遍，通过对比脚本的方法来查找需要关联的部分，但并不是说两次录制的所有不同点都需要关联，这个要具体情况具体分析。

1. 录制登陆CRM系统并新增一个客户的脚本

双击打开CRM被测系统文件夹下的专业调试工具.exe，然后打开PR，新建一个项目，在Action.bsh脚本文件夹下点击开始录制按钮，填好参数之后，点击确定。

1. 进行关联

点击菜单栏执行-关联，出现需要关联的脚本，点击确定，若关联尚未完成会弹出“还没有做完所有关联，请继续进行关联操作”重复第一步，直到出现“关联结束，无需进一步关联”的消息框。

（3）运行脚本。

点击执行按钮，开始执行，若执行失败，把失败对应的脚本段删除，或者重新录制。直到执行成功。

**2.5脚本参数化**

一条普通脚本只能执行某个特定的动作，若想执行另一个动作还得把脚本手动做修改，但是这样做显然不符合自动化的理念，为了解决这个问题，我们引入“参数化”的概念，将脚本参数化后，不需要每次都更改脚本语句就可以执行不同的功能。脚本参数化之前，必需要编辑好参数表。

1. 录制登陆再退出被测系统CRM的脚本

程序输入参数填写CRM系统的地址，录制之前还需先启动CRM系统，然后点击确定，开始录制。输入用户名、密码，点击登录。

1. 编辑参数表

把username和password部分的脚本进行参数化，然后设置好参数表。

1. 循环参数表

输入对应的多组用户名、密码

（4）运行脚本查看参数化结果

运行脚本，查看信息，看是否循环参数表成功。

**2.6事物**

录制的脚本没有插入事务时，是一个整体，很难分析系统瓶颈是哪些动作导致的，所以ＰＲ引入事务。可以把一个较大的脚本中不同的动作，分成不同的事务，然后进入性能分析,这样在性能分析里边就会把每个事务分别进行分析。这样就可以更详细，具体地知道是用户的哪些动作对系统性能影响比较大。

事务：就是在脚本中定义的某段操作，也可以说是一段脚本语句。

1. 录制打开CRM登陆界面的脚本

（２）进入执行器新建场景，场景下添加项目

（３） 脚本中添加事务，有开始有结束，添加事物后执行脚本，查看结果

**2.7集合点**

集合点用以同步虚拟用户以便恰好在同一时刻执行任务。在测试计划中，可能会要求系统能够承受100 人同时登录，在PR中可以通过在登录操作前面加入集合点，这样当虚拟用户运行到登录的集合点时，PR就会检查同时有多少用户运行到集合点，如果不到50人，PR就会命令已经到集合点的用户在此等待，当在集合点等待的用户达到50 人时，PR命令50人同时去登录，从而达到测试计划中的需求。

假设创建场景时设置了50个虚拟用户同时加载，也就是所谓的50个用户同时并发，那么这个并发过程是不是能够贯穿整个运行过程呢？答案是否定的，客观的讲，这个并发过程仅仅体现在开始执行的一刹那，随着服务器对请求的响应时间的不一致或系统环境条件的限制，50个用户的步调就会马上表现出不一致，有的快些，有的慢一些，在运行的过程中能够集合到一个点上的可能性微乎其微。

　　系统负荷最大的情况是所有用户都集中到系统瓶颈的某个点上进行操作。为了解决这个情况，PR提供了集合点的功能，帮助测试人员实现真正意义上的并发。

（１）录制CRM脚本并回放。

（２）添加集合点。

在需要添加集合点的地方，添加如下脚本语句即可pr.rendezvous("login");括号内为集合点名字，可以随意取名。

虚拟用户执行到集合点处时，暂时的停留，直到虚拟用户数满足我们的测试需求时，才执行集合点之后的脚本，达到同时并发，测试系统承受能力的目的。

1. 插入集合点，添加完成后脚本能执行成功

**2.8 IP地址欺骗**

当运行Scenario时，Vuser使用PR所在机器的固定IP地址，同时每个PR上运行大量的虚拟用户，这样就造成了大量的用户使用同一IP同时访问一个网站的情况，这种情况和实际运行的情况不符，并且有一些网站在设计的时候会根据用户IP来分配资源，这些网站会限制同一个IP的多次登陆。为了更加真实的模拟实际环境，PR允许运行的虚拟用户使用不同的IP访问同一网站,这种技术称为“IP 欺骗”。

启用该技术后，场景中运行的虚拟用户将模拟从不同的IP 地址向网站发送请求，从而避免了网站限制登录的情况。

（１）打开PR，新建项目

（２）新建场景，并与项目相关联

（３）IP欺骗的设置

进入运行界面，点击"IP欺骗设置"，启用IP欺骗前打勾，然后点击"IP欺骗"；本地IP设置

**2.9场景设置**

要使用 PerformanceRunner 测试被测系统，必须创建负载测试场景。场景定义每次测试期间发生的事件。场景定义并控制要模拟的用户数，这些用户执行的操作以及用于运行模拟场景的计算机。执行器负责设计场景、运行场景、控制场景、各种波形图生成等。

（１）新建项目，录制CRM系统登陆退出脚本。

（２）在执行器中新建场景，并在场景中添加项目。

（３）启动组设置。

启动组，设定了场景何时启动，可以选择立即启动，也可以选择在等待一段时间后再启动，或者是在运行完某一个场景后启动（用在执行多场景的时候）。

（４）启动虚拟用户设置。

启动虚拟用户，该项设定了所有虚拟用户的启动方式，可以选择所有虚拟用户同时启动，也可以选择每个指定时间内启动指定数目的虚拟用户。

（５）持续时间设置。

持续时间，该项设定了所有虚拟用户在启动之后运行的时间，可以选择执行完测试脚本后立即停止，也可选择持续运行一段时间后停止。

（６）停止虚拟用户设置。

停止虚拟用户，该项设定了所有虚拟用户的停止方式，可以选择所有虚拟用户同时停止，也可以选择每个指定时间内停止指定数目的虚拟用户。

（７）运行

**2.10报告分析**

运行的vuser图：对于判断在给定的时间点被测对象上的虚拟用户负载非常有用。另外此图常常跟其他的图表进行关联，比如跟事务响应时间图、吞吐量图等等进行关联，可以看出在多大用户量的情况下响应时间最大，吞吐量最高等等

事务概要图：显示了场景或会话步骤中成功的事务和失败的事务的最小、最大和平均性能时间

事务响应时间：是在测试场景运行期间的每一秒内，执行各个事务所用的平均时间，通过它可以分析测试场景运行期间应用系统的性能走向。此外，事务平均响应时间图还提供了测试场景运行期间内各个事务响应时间的最大值、最小值、平均值等信息，这些值是度量事务响应时间是否满足用户需求的重要参考依据

每秒事务数：（TRS）是在场景运行的每一秒中，每个事务通过、失败以及停止的数量，是考查系统性能的一个重要参数，通过它可以确定系统在任何给定时刻的实际事务负载，通过分析单位时间内通过的事务数，可以直接看出系统的性能变化趋势

每秒事务总数：是场景在运行时，每一秒内通过的事务总数、失败的事务总数以及 停止的事务总数。如果西性能稳定，在同等压力下，每秒通过事务总数图应该接近一条直线，而不是逐渐倾斜。与每秒通过事务数（TRS）图相比，每秒通过事务总数图更关注服务器整体处理事务的情况，是一个宏观的概念

事务性能概要图：查看事务中每个页面组件的下载时间. 显示了场景或会话步骤中所有事务的最小、最大和平均性能时间

每秒点击量：每秒点击次数即点击率图，是指在场景运行过程中虚拟用户每秒向Web服务器提交的HTTP请求数

吞吐量（字节/兆）：是指在场景运行过程中服务器每秒的吞吐量。表示在任何给定的某一秒钟虚拟用户从服务器获得的数据量。依据服务器的吞吐量可以评估虚拟用户产生的负载量

CPU使用率：显示单位时间内占用CPU资源的情况

物理内存使用：单位时间内程序占用内存的百分率

网络流量：显示单位时间内的网络流量（上传与下载）

**3 小结**

压力测试的关键是场景设置，报告分析，整合测试结果数据，分析出系统性能行为表现的规律，并准确定位系统的性能瓶颈所在。在这个步骤里，可以利用数学手段对大批量数据进行计算和统计，使结果更加具有客观性。在压力测试中，需要注意的是，能够执行的压力测试方案并不一定是成功的，成败的关键在于其是否精确地对真实世界进行了模拟。

在整个压力测试过程中，自动化测试工具的选择只能影响压力测试执行的复杂程度，简便一些或繁杂一些；但人的分析和思考却会直接导致压力测试的成败。所以要着重于对压力测试思路的整理。

**４ 思考题**

1. 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？
2. PR的主要功能和使用流程？
3. PR的主要元件？
4. 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？
5. 尝试自学JMeter，并与PR全面比较。