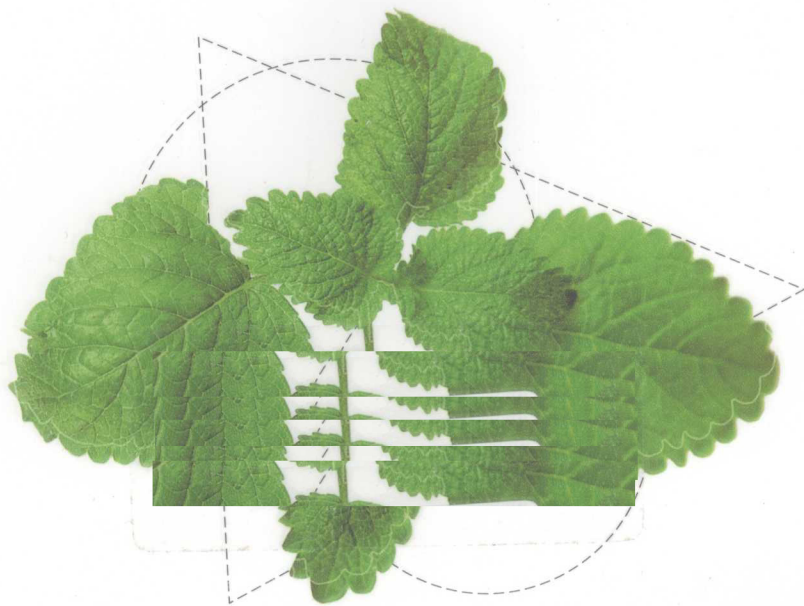


测试，还是测试！
我们介绍的不仅仅是工具，还有思想！



LoadRunner

性能测试完全讲义

黄文高 赵丹 编著

- 资深性能测试工程师深入剖析LoadRunner每个功能的“内幕”
- 全程实战，让你遨游于性能测试世界，体会性能测试全过程
- 超长语音视频讲解，教你如何玩转LoadRunner，同作者一起“悟道”
- 性能测试交流论坛（www.testingba.com）即将推出，敬请期待……



中国水利水电出版社
www.waterpub.com.cn

LoadRunner 性能测试完全讲义

黄文高 赵丹 编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

在软件测试行业,性能测试和自动化测试成为初级软件测试工程师迈向高级测试工程师必须跨越的一道门槛,而 LoadRunner 是性能测试工具的一面旗帜,谈到性能测试就不能不谈到 LoadRunner。

本书分三部分:入门篇、提高篇和实战篇。入门篇主要介绍性能测试基础知识、LoadRunner 基础知识和 LoadRunner 三大组件;提高篇采用大量的实例介绍 LoadRunner 的功能;实战篇使用两个案例就如何使用 LoadRunner 进行性能测试做了详细的介绍,从需求分析到结果分析都进行了详细的讲解,旨在通过案例分析功能或介绍使用技巧,希望读者能理解案例解决方案背后的思考过程、分析过程和推导过程。

本书适合暂时不了解性能测试,但又想了解性能测试、学习 LoadRunner 的读者;适合 LoadRunner 的初学者,希望看完本书他们能有很大的提高;适合中级性能测试工程师,希望本书的实践对他们的工作有益。

本书配套光盘中提供了书中实例所用脚本文件,PPT 电子讲义,以及书中内容的详细视频讲解,405 分钟超长视频与本书内容完美结合,深化 LoadRunner 的重点与难点,详细解剖 LoadRunner 的每个功能,带您轻松步入性能测试之路。

图书在版编目(CIP)数据

LoadRunner性能测试完全讲义 / 黄文高, 赵丹编著
— 北京: 中国水利水电出版社, 2010. 5
ISBN 978-7-5084-7457-1

I. ①L… II. ①黄… ②赵… III. ①性能试验—软件工具, LoadRunner IV. ①TP311.56

中国版本图书馆CIP数据核字(2010)第075040号

策划编辑: 周春元 责任编辑: 李 炎 封面设计: 李 佳

书 名	LoadRunner 性能测试完全讲义
作 者	黄文高 赵丹 编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×240mm 16 开本 17 印张 403 千字
版 次	2010 年 5 月第 1 版 2010 年 5 月第 1 次印刷
印 数	0001—4000 册
定 价	38.00 元 (赠 1DVD)

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前言

目前,在国内,软件测试尚处于起步阶段,其市场表现实在有点令人尴尬。软件开发公司比比皆是,但软件测试公司却如凤毛麟角。市场化的第三方测试如同刚刚出生的宁馨儿,目前的市场几乎可以忽略不计。

国内IT企业也逐步开始重视对软件测试团队的建设,一些知名IT企业已经将软件测试作为企业未来发展的一个版块。目前在国内软件测试行业中,各种软件测试的方法、技术和标准都还在探索阶段。

有媒体报道过,国内开发人员与测试人员的比例是8:1,而国际公认的行业标准应该是1:1,可见国内测试行业与国外的差距比较大,这说明软件测试行业未来发展前景一片光明。

但是,从长远发展角度来看,测试还是需要高端人才。自动化测试、性能测试和对Bug的预防及早期检查是软件测试工程师面临的新的挑战。值得庆幸的是,近几年国内一些IT企业已经开始涉及自动化测试和性能测试。在软件测试行业,性能测试和自动化测试成为初级软件测试工程师迈向高级测试工程师必须跨越的一道门槛,随便打开一个软件测试的网站,都能看到性能测试的版块,而LoadRunner是性能测试工具的一面旗帜,谈到性能测试就不能不谈到LoadRunner。

不论是性能测试,还是性能测试工具的杰出代表——LoadRunner,许多朋友在现实工作中并没有真正掌握。虽然会使用LoadRunner进行性能测试,但是测试结果并不能令人满意。笔者在工作中发现许多使用LoadRunner的人其实并不真正理解性能测试的意义,许多朋友不清楚如何分析性能测试需求?什么叫业务模型?什么叫场景模型?如何分析测试结果?笔者认为读者不应该仅仅满足于学会使用LoadRunner,而应该能够从学习LoadRunner的过程中“悟道”。

从全国软件测试培训机构就LoadRunner培训内容调查,读者朋友最希望了解的内容是制定性能测试计划方案,进行性能测试需求分析、测试脚本开发、场景设计、测试执行及结果分析。故本书设计了两个不同架构的案例来分析使用LoadRunner进行性能测试的整个过程,不仅仅是讲述使用LoadRunner如何进行性能测试,更重要的目的是借案例来分析性能测试的思想,工具仅仅是载体,而思想才是最重要的。

本书结构清晰,内容安排由浅入深,对初学者来说可以很轻松的入门,当然对于有经验的读者朋友来说,可以有选择性的阅读。主要包括入门篇、提高篇和实战篇。

入门篇主要介绍的内容有:性能测试基础知识、LoadRunner基础知识和LoadRunner三大组件。性能测试基础知识部分主要介绍了性能测试过程中一些常见的术语、性能测试划分和性能测试应用的领域;LoadRunner基础知识部分主要介绍了LoadRunner的工作原理、工作过程和的内部结构,从全局的角度对LoadRunner工具进行了介绍;LoadRunner三大组件部分主要介绍了Vuser发生器、Controller控制器和Analysis分析器。在入门篇中只是简单介绍三大组件的常用操作及工作原理。

在提高篇中，将对 VuGen、Controller 控制器和 Analysis 分析器三大组件进行深入的了解，通过实例对测试过程中常用的技巧与方法进行详细的分析，就测试过程中容易出现错误的地方进行提示。

在实战篇中，选择两个案例就如何使用 LoadRunner 进行性能测试的整个过程进行全面的介绍。选择的两个实例的系统架构模式分别是 B/S 模式和 C/S 模式，目的是通过选取不同架构的系统进行实验。在实验过程中就这两种架构的系统在性能测试过程中的不同之处给予详细说明，并尽可能地在实战过程中引出更多的问题，希望给初学者以更多的帮助。

本书采用大量的实例和脚本对 LoadRunner 的使用进行介绍，从提高篇开始，本书就使用了大量的案例，目的是通过案例分析测试过程中常用的技巧，并且就测试过程中容易出错的地方进行提示，让读者朋友不仅仅是在看书，更重要的是在动脑思考。实战篇介绍的不仅仅是如何使用 LoadRunner，更重要的是分析性能测试的整个过程。最后两章使用两个案例讲解如何使用 LoadRunner 进行性能测试，从需求分析到结果分析都进行了详细的介绍。

本书中某些实例或许正好与读者面临的问题相同或存在相似之处，读者可以依葫芦画瓢地去解决现实中的问题，然而这并非作者的本意。作者的初衷是将思考方法和分析过程通过实例的方式传达给读者并与读者进行交流，希望读者能理解案例解决方案背后的思考过程、分析过程和推导过程。如果读者经过思考得出与作者不同的分析结果，或是证明作者所给出的解决方案并非最好的方案，这也是作者所期望的。

笔者希望读者在阅读本书的过程中，认真思考案例中如何进行需求分析、如何建立业务模型、如何建立场景模型和分析测试结果的过程，并将这些方法应用到现实的工作中去，而不要沉迷于书中给出的具体案例。另外，作者的分析方法和推导过程只是作者本人在工作中自己总结出的经验，不是标准答案，更不是圣经。期望读者能够认真思考作者的这些经验，并结合自己的实际，总结出一套自己的方法。如果真是这样，作者的这些文字工作就真正劳有所值了。

经过一年多的努力，书稿终于完成，在这里我感谢那些曾经帮助、支持和鼓励过我的朋友。感谢中国水利水电出版社周春元编辑的帮助。

感谢我的同事黄胜杰、陈志坚、梁会美、农莉、沈东雪、向小飞、白晓霞和我的朋友黄金、黄海平、包静、万俊杰帮忙审核书稿。

感谢我的妻子韦玉凤奉献的爱心、支持与鼓励，并将此书献给我即将出生的孩子。

感谢父亲和母亲这些年来对我的养育之恩。

由于笔者水平有限，很多内容是自己的经验总结，出现错误在所难免，欢迎广大读者批评指正。读者在阅读本书的过程中如有任何不清楚的问题和批评建议，可以发邮件到 huangwengao@sina.com，作者将尽力给您答疑解惑。

最后，感谢您购买此书，希望您在本书中能够找到那些正在困扰着您的问题的答案。祝大家阅读愉快。

黄文高
2010年2月

目 录

前言

第一部分 入门篇

第1章 性能测试基础知识	2	2.5 LoadRunner 测试步骤	16
1.1 什么是软件的性能	2	第3章 Vuser 发生器	18
1.2 性能测试相关术语	3	3.1 脚本录制	18
1.2.1 响应时间	3	3.1.1 如何选择协议	19
1.2.2 并发用户数	4	3.1.2 开始录制脚本	22
1.2.3 吞吐量	4	3.2 Recording Options 设置	24
1.2.4 吞吐率	5	3.2.1 Recording 选项卡	24
1.2.5 TPS	5	3.2.2 Advanced 选项卡	26
1.2.6 点击率	5	3.2.3 Correlation 选项卡	27
1.2.7 资源利用率	5	3.3 Run-Time Settings 设置	27
1.2.8 性能计数器	6	3.3.1 Run Logic 选项卡	28
1.2.9 思考时间	6	3.3.2 Pacing 选项卡	29
1.3 性能测试划分	6	3.3.3 Think Time 选项卡	29
1.3.1 负载测试	7	3.3.4 Miscellaneous 选项卡	30
1.3.2 压力测试	7	3.4 脚本完善	31
1.3.3 配置测试	7	3.4.1 插入事务	31
1.3.4 并发测试	7	3.4.2 插入集合点	33
1.3.5 可靠性测试	8	3.4.3 插入注释	34
1.4 性能测试应用领域	8	第4章 Controller 控制器	36
1.4.1 能力验证	8	4.1 场景类型介绍	36
1.4.2 规划能力	8	4.1.1 手动测试场景	36
1.4.3 性能调优	9	4.1.2 面向目标测试场景	38
1.4.4 缺陷发现	9	4.2 场景设计	39
第2章 LoadRunner 基础知识	10	4.2.1 手动场景 Schedule 配置	39
2.1 LoadRunner 简介	10	4.2.2 面向目标场景 Schedule 配置	43
2.2 LoadRunner 工作原理	11	4.2.3 配置 View Script	47
2.3 LoadRunner 工作过程	12	4.2.4 配置 Load Generator	47
2.4 LoadRunner 内部结构	13	4.3 场景执行	48

4.3.1 场景控制	48	5.2.2 统计部分	75
4.3.2 场景执行期间查看场景	53	5.2.3 事务统计部分	76
4.4 场景监视	57	5.2.4 HTTP 响应统计	77
4.4.1 关于联机监控	57	5.3 Analysis 常见图分析	79
4.4.2 监控器与度量	58	5.3.1 Vuser 图	79
4.4.3 联机监视器	62	5.3.2 每秒点击数图	80
第 5 章 Analysis 分析器	66	5.3.3 平均事务响应时间图	80
5.1 Analysis 简介	66	5.3.4 吞吐量图	81
5.1.1 Analysis 基础知识	66	5.4 Analysis 报告	82
5.1.2 设置选项	67	5.4.1 HTML 报告	82
5.1.3 Analysis 图	72	5.4.2 Word 报告	82
5.2 摘要报告	74	5.4.3 水晶报表	85
5.2.1 概要部分	75		

第二部分 提高篇

第 6 章 脚本编写	92	7.4 执行路径转换	138
6.1 检查点	92	7.4.1 路径转换介绍	138
6.1.1 插入检查点	93	7.4.2 编辑路径转换表	139
6.1.2 检查点函数	99	7.5 在 LoadRunner 中使用功能测试脚本	141
6.2 Block (块) 技术	101	7.5.1 QuickTest 创建 GUI Vuser 脚本	142
6.3 参数化技术	106	7.5.2 WinRunner 创建 GUI Vuser 脚本	143
6.3.1 创建参数	106	7.5.3 场景中使用 GUI Vuser 脚本	144
6.3.2 参数类型属性	108	第 8 章 结果分析实践	146
6.3.3 数据文件	108	8.1 分析图合并	146
6.3.4 导入数据	114	8.1.1 分析图合并原理	146
6.4 关联技术	118	8.1.2 实例讲解	148
6.4.1 录制中关联	119	8.2 分析图关联	150
6.4.2 录制后关联	121	8.2.1 分析图关联原理	150
6.4.3 手动关联	123	8.2.2 实例讲解	152
第 7 章 场景设计实践	127	8.3 页面细分	154
7.1 集合点设置	127	8.3.1 页面细分原理	154
7.2 IP 欺骗技术	130	8.3.2 实例讲解	157
7.2.1 IP Spoofer 设置	131	8.4 钻取技术	159
7.2.2 Controller 中启动 IP Spoofer	134	8.4.1 钻取技术原理	159
7.3 负载均衡技术	136	8.4.2 实例讲解	160

8.5 导入外部数据	161
8.5.1 导入数据工具	161
8.5.2 自定义文件格式	164
第9章 特殊协议	167
9.1 Windows Sockets (WinSock) 协议	167
9.1.1 Windows Sockets 录制选项设置	168
9.1.2 Windows Sockets 录制	169

9.1.3 Windows Sockets 数据操作	173
9.1.4 关于 LRS 函数	177
9.2 邮件服务协议	182
9.2.1 邮件服务协议简介	182
9.2.2 邮件服务协议录制	184
9.2.3 脚本分析	190
9.2.4 关于 SMTP 和 POP3 函数	192

第三部分 实战篇

第10章 客户关系管理系统性能测试	196
10.1 系统介绍	196
10.2 需求分析	198
10.2.1 性能指标	198
10.2.2 需求详细分析	198
10.3 测试方案及计划	200
10.3.1 人力资源	200
10.3.2 时间进度	200
10.3.3 测试环境准备	200
10.3.4 业务模型创建	201
10.3.5 场景模型创建	202
10.3.6 测试数据准备	203
10.4 测试用例	204
10.5 执行测试	207
10.5.1 脚本开发	207
10.5.2 场景设计	212
10.5.3 计数器设置	217
10.5.4 场景监视	219
10.6 结果分析	221
10.7 测试结论	229
第11章 信息系统性能测试	230

11.1 系统介绍	230
11.2 需求分析	231
11.2.1 性能指标	231
11.2.2 需求详细分析	231
11.3 测试方案及计划	232
11.3.1 人力资源	232
11.3.2 时间进度	232
11.3.3 测试环境准备	233
11.3.4 业务模型创建	234
11.3.5 场景模型创建	234
11.3.6 测试数据准备	235
11.4 测试用例	236
11.5 执行测试	237
11.5.1 脚本开发	237
11.5.2 场景设计	241
11.5.3 计数器设置	247
11.5.4 场景监控	247
11.6 结果分析	249
11.7 测试结论	257
附录A 主要计数器	258
附录B 性能测试 i 模型	262

第一部分

入门篇

入门篇主要介绍的内容有：性能测试基础知识、LoadRunner 基础知识和 LoadRunner 的三大组件。性能测试基础知识部分主要介绍了性能测试过程中一些常见的术语、性能测试划分和性能测试应用的领域；LoadRunner 基础知识部分主要介绍了 LoadRunner 的工作原理、工作过程和内部结构，从全局的角度对 LoadRunner 工具进行介绍；LoadRunner 的三大组件部分主要介绍了 Vuser 发生器、Controller 控制器和 Analysis 分析器。入门篇中只是简单介绍三大组件的常用操作及工作原理，其使用技巧和高级使用将在提高篇中进行详细的介绍。

性能测试基础知识

1.1 什么是软件的性能

一般来说，性能是一种指标，表明软件系统或构件对其及时性要求的符合程度；其次，性能是软件产品的一种特性，可以用时间来进行度量。

性能的及时性用响应时间或吞吐量来衡量。响应时间是指服务器对请求作出响应所需要的时间。

对于单个事务，响应时间是指完成事务所需的时间；对于用户任务，响应时间体现为端到端的时间。比如，“用户点击‘确定’按钮后 3 秒内呈现出结果”就是一个对用户任务响应时间的描述，在这个用户任务中，可能有多个具体的事务需要完成，每个事务都有其单独的响应时间。

对交互式应用（如 Web 应用）来说，一般以用户感受到的响应时间来描述系统的性能，而对非交互式应用（如嵌入式系统或银行等业务处理）而言，响应时间是指系统事件产生响应所需的时间。

通常，关注软件性能的对象是多个层面的，用户关注软件性能，系统管理员关注软件性能，软件开发工程师也关注软件性能，下面从三个不同的层面对软件性能做一个简要的介绍。

1. 用户

从用户的角度来说，软件性能是软件对用户操作的响应时间。通俗的讲，如果用户点击一个提交或输入一个 URL 地址，随后系统把结果呈现到用户眼前，这个过程所花费的时间即为用户对软件性能的直观印象，如图 1-1 所示。

需要注意的一点是，用户体会到的“响应时间”，既有客观的成分，也有主观的成分，从提交业务到系统开始返回信息的时间被用户认为是系统的响应时间，例如，用户执行某个操作，该操作会返回大量的数据，假如所有的数据并不会同时返回，而是先将一部分数据呈现出来，再将全部数据呈现出来，此时，用户体会的响应时间为从执行操作到有部分数据呈现出来的时间，而真正的响应时间应该是系统将全部数据呈现出来的时间。

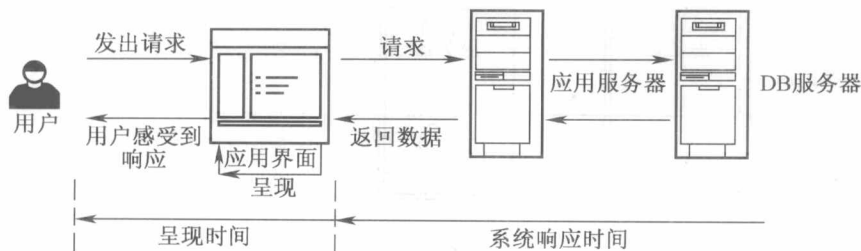


图 1-1 Web 系统响应

2. 系统管理员

从系统管理员的角度来说,软件性能在响应时间方面与用户的视角是一样的。但管理员是一群特殊的用户群体,管理员除关注一般用户体验外,还关注与系统状态相关的信息。如系统资源的使用情况,包括 CPU 的使用情况、内存的使用情况、磁盘 I/O 等,当然还有数据交互的情况。

另外,管理员还关注系统硬件资源的可扩展性即规划性能部分。比如,系统现在支持 100 个用户并发没问题,那么将来支持 200 个用户并发时是否会出现性能问题呢?

3. 软件开发工程师

从软件开发工程师的角度来说,他们关注用户和管理员关注的所有问题。另外还关注内存泄漏、数据库是否出现死锁、中间件以及应用服务器等问题。

1.2 性能测试相关术语

本小节将介绍性能测试过程中的一些相关术语:响应时间、并发用户数、事务响应时间、吞吐量、吞吐率、TPS(每秒事务响应数)、性能计数器等。

1.2.1 响应时间

响应时间是指应用系统从发出请求开始到客户端接收到所有数据所消耗的时间。该定义强调所有数据都已经被呈现到客户端所花费的时间,为什么说是所有数据呢?因为用户体验的响应时间带有主观性,用户认为从提交请求到服务器开始返回数据到客户端的这段时间为响应时间。

现在对响应时间进行细分,以一个 Web 应用的页面响应时间为例,如图 1-2 所示。从图中可以看到,页面的响应时间可分解为“网络传输时间”(N1+N2+N3+N4)和“应用延迟时间”(A1+A2+A3),而“应用延迟时间”又可分解为“数据库延迟时间”(A2)和“应用服务器延迟时间”(A1+A3)。

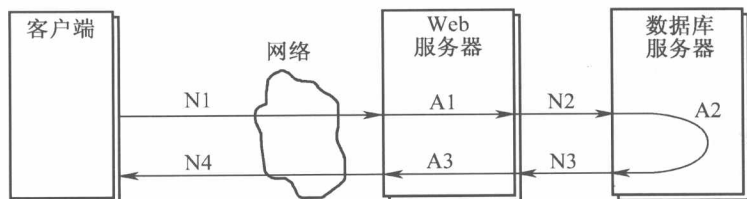


图 1-2 Web 页面响应时间分解

1.2.2 并发用户数

并发用户数指同一时刻与服务器进行数据交互的所有用户数量。概念中有两点需要注意。第一：同一时刻，因为并发强调的是用户同时对服务器进行施压。例如：一个人同时挑两件东西，这时表示两件东西同时被这个人挑起来，而如果是先挑一件，再挑另外一件，那么就无法表现出同时的概念，这两件东西也就没有同时施压在这个人身上。第二：强调要与服务器进行数据交互，如果未和服务器进行数据的交互，这样的用户是没给服务器带来压力的。同样是上面的例子，这个人虽然同时挑了两件东西，但其中有一件东西是没重量的，那就是说只有一件东西对这个人造成压力。

因此对于并发用户这个概念的理解经常会出现以下两种误区：一是认为系统所有的用户都叫并发用户；二是认为所有在线的用户都是并发用户。在线用户不一定是并发用户，原因是在线用户不一定就与系统进行了数据的交互，例如：如果一些在线用户只是查看系统上的一些消息，那么这些在线用户不能作为并发用户计算，因为这些用户并没有与系统进行数据交互，不会给服务器带来任何压力。

那么并发用户数如何计算呢？目前并没有一个精确的计算公式，很多情况下都是根据以往的经验进行估算。根据行业的不同，并发用户数也会有所不同，像电信行业并发用户数为在线用户的万分之一，如果有 1000 万在线用户，那么需要测试 1000 个并发用户即可。OA（办公自动化）系统的并发用户数一般是在线用户的 5%~20% 左右，所以并发用户数很大程度上是根据经验和行业的一些标准来计算的。

1.2.3 吞吐量

在性能测试过程中，吞吐量是指单位时间内服务器处理客户请求的数量，吞吐量通常使用请求数/秒来衡量，其直接体现服务器的承载能力。

吞吐量作为性能测试过程中主要关注的指标之一，它与虚拟用户数之间存在一定的联系，当系统没有遇到性能瓶颈时，可以采用下面这个公式来计算。

$$F = \frac{N_{vu} \times R}{T}$$

其中，F 表示吞吐量； N_{vu} 表示 VU（Virtual User，虚拟用户数）的个数；R 表示

每个 VU 发出的请求数量；T 表示性能测试所用的时间。但是如果系统遇到性能瓶颈，这个公式就

不再适用。吞吐量与 VU 之间的关联图如图 1-3 所示。从图中可以看出, 吞吐量在 VU 增长到一定数量时, 软件系统出现性能瓶颈, 此时, 吞吐量的值并不会随着 VU 数量的增加而增大, 而是趋于平衡。

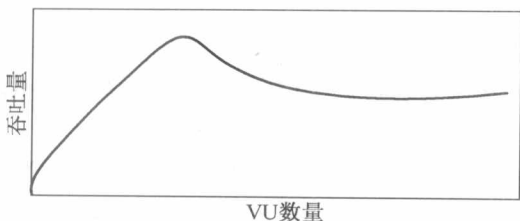


图 1-3 吞吐量-VU 数量关系图

1.2.4 吞吐率

吞吐率 (Throughput) 是指单位时间内从服务器返回的字节数, 也可以指单位时间内服务器处理客户提交的请求数。它是衡量网络性能的一个重要指标。吞吐率=吞吐量/测试时间, 通常情况下吞吐量的值越大, 吞吐率的值也越大, 吞吐率的值越大系统的负载能力越强。

1.2.5 TPS

TPS (Transaction Per Second) 表示服务器每秒处理的事务数, 它是衡量系统处理能力的重要指标。

1.2.6 点击率

点击率 (Hit Per Second) 是指每秒钟用户向服务器提交的 HTTP 数量。用户每点击一次, 服务器端就要对用户提交的请求进行一次处理, 从事务的角度来说, 如果把每次点击作为一次提交事务来对待, 那么点击率与 TPS 的概念是等同的。对于 Web 系统来说, “点击率” 是服务器处理的最小单位, 点击率的值越大, 说明服务器端所能承受的压力越大。因此通常情况下, Web 服务器都具有防刷新的机制, 因为客户每刷新一次系统就要响应一次点击, 如果不对服务器进行防刷新处理, 当用户不停地点击刷新按钮, 此时服务器将承受巨大的压力。

需要注意的是, 点击一次并不代表客户端只向服务器端发送一个 HTTP 请求, 客户每点击一次, 都会向服务器端发出多个 HTTP 请求。

1.2.7 资源利用率

资源利用率是指服务器系统中不同硬件资源被使用的程度, 资源使用率=资源实际使用量/总的

可用资源量。主要包括 CPU 利用率、内存利用率、磁盘利用率、网络等。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,在配置调优测试的过程中,通过比较配置调优前后系统资源的利用率来判断调优的效果。

1.2.8 性能计数器

性能计数器(Counter)是描述服务器或操作系统性能的一些数据指标。主要是通过添加计数器来观察系统资源的使用情况。性能计数器包括操作系统性能计数器、数据库计数器、应用服务器计数器等。

计数器在性能测试过程中发挥着“监控和分析”的关键作用,尤其是在分析系统的可扩展性和对性能瓶颈进行定位时,计数器的阈值起着非常重要的作用。必须注意的是,一般情况下,单一的性能计数器只能体现系统性能的某一个方面,在性能测试过程中分析测试结果时,必须基于多个不同的计数器进行分析。

在性能测试中常用资源利用率进行横向对比。如在进行性能测试时发现,某个资源的使用率很高,几乎达到 100%,假设该资源是 CPU,而其他资源的使用率又比较低,这时可以很清楚地知道,CPU 是系统性能的瓶颈。

1.2.9 思考时间

思考时间(Think Time),也称为“休眠时间”,是指用户在进行操作时,每个请求之间的时间间隔。对于交互系统来说,用户不可能持续不断地发出请求,一般情况下,用户在向服务器端发送一个请求后,会等待一段时间再发送下一个请求。

在测试脚本中,思考时间为脚本中两条请求语句之间的间隔时间。当前对于不同的性能测试工具提供了不同的函数来实现思考时间,在实际的测试过程中,如何设置思考时间是性能测试工程师要关心的问题。

1.3 性能测试划分

性能测试划分有很多种,测试方法也有很多种,更确切的说是由于测试方法的不同决定了测试划分的情况,但在测试过程中性能测试的划分没有绝对的界限,常用的有压力测试、负载测试和并发用户测试等。

性能测试的方法主要包括以下几种:

- 负载测试 (Load Testing)
- 压力测试 (Stress Testing)
- 配置测试 (Configuration Testing)
- 并发测试 (Concurrency Testing)
- 可靠性测试 (Reliability Testing)

1.3.1 负载测试

负载测试 (Load Testing) 是通过对被测试系统不断地加压, 直到超过预定的指标或者部分资源已经达到了一种饱和状态不能再加压为止。就像举重运动员, 在举重的过程中不断地增加杠铃重量, 直到运动员无法举起。

该方法主要是为了找到系统最大的负载能力, 为性能调优提供数据。该测试方法有以下几个特点:

- 1) 目的: 找到系统最大的负载能力。
- 2) 环境: 该方法需要在特定的环境下进行测试。
- 3) 手段: 不断地对系统进行加压, 直到系统中部分资源达到极限。

1.3.2 压力测试

压力测试 (Stress Testing) 是指系统已经达到一定的饱和程度 (如 CPU、磁盘等已经处于饱和状态), 此时系统处理业务的能力, 系统是否会出现错误。

疲劳测试是压力测试的一种表现形式。例如, 一个人很累了, 但还在持续不停的工作。

该测试方法有以下几个特点:

- 1) 目的: 测试在系统已经达到一定的饱和程度时, 系统处理业务的能力。
- 2) 手段: 使用模拟负载等方法, 使系统资源达到一个较高的水平。
- 3) 该方法一般用于系统稳定性测试。

1.3.3 配置测试

配置测试 (Configuration Testing) 是通过调整系统软/硬件环境, 了解各种不同环境对系统性能的影响, 从而找到系统的最优配置。

该测试方法有以下几个特点:

- 1) 目的: 通过调整环境了解不同因素对系统性能的影响情况, 从而找到调优的方法。
- 2) 手段: 通过调整系统软/硬件环境, 使系统在不同环境下进行性能测试。
- 3) 该方法一般用于系统调优和规划能力。

1.3.4 并发测试

并发测试 (Concurrency Testing) 是通过模拟用户并发访问, 测试多用户同时访问同一应用、模块或数据, 观察系统是否存在死锁、系统处理速度是否明显下降等其他的一些性能问题。

该测试方法有以下几个特点:

- 1) 目的: 当多用户并发访问时, 系统是否存在一些可能的并发问题。
- 2) 手段: 模拟多用户同时并发操作。

1.3.5 可靠性测试

可靠性测试 (Reliability Testing) 是当系统在一定的业务压力下, 让系统持续运行一段时间, 观察系统是否达到要求的稳定性, 此处强调在一定业务压力下持续运行的能力, 可靠性测试必须给出一个明确的要求, 如系统能够持续无故障运行多少天。

该测试方法有以下几个特点:

- 1) 目的: 测试系统在一定的业务压力下, 系统可持续运行的时间。
- 2) 环境: 指明系统在一定的业务压力环境下持续运行。
- 3) 测试过程中要关注系统运行的情况。

1.4 性能测试应用领域

上一小节讲了常用的性能测试方法, 本小节将从性能测试的应用领域来讲述性能测试的分类, 从应用领域来划分, 性能测试分为以下四大领域:

- 能力验证
- 规划能力
- 性能调优
- 缺陷发现

1.4.1 能力验证

能力验证是性能测试最常用的一个领域。一般能力验证采用这样的描述方式: “某系统能否在条件 A 下具备 B 性能”。重点在于验证系统是否具备某种能力。

能力验证领域有以下几个特点:

- 1) 要求在一个已确定的环境下运行。
- 2) 需要根据典型场景来设置测试方案与测试用例。

1.4.2 规划能力

规划能力与能力验证有相似之处, 但还是存在一些不同的地方, 能力验证强调的是在某个条件下具备什么样的能力, 而规划能力体现系统如何才能达到要求的性能指标。规划能力问题常常会这样描述: “系统如何才能支持未来用户增长的需要”, 这里强调的是未来能力增长的一个需求, 着眼于未来系统的规划。

规划能力领域的特点是:

- 1) 对系统能力的一种探索性的测试。
- 2) 可以了解系统的性能及系统性能的可扩展性。

1.4.3 性能调优

性能调优是通过测试来调整系统的环境,最终使系统性能达到最优的状态。这是一个持续调优的过程,主要调优的对象有数据参数、应用服务器、系统的硬件资源等。

一个标准性能调优的步骤如图 1-4 所示。

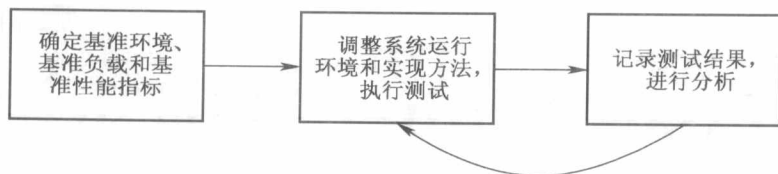


图 1-4 性能调优过程

1.4.4 缺陷发现

性能测试应用领域的主要目标是通过性能测试的手段来发现系统存在的缺陷。很多系统在实验室测试环境中没有任何问题,可是当交付给客户时就出现了莫名其妙的错误。如果交付给客户后出现多人同时访问速度缓慢或宕机的现象,那么很有可能是由于系统性能问题所引起。