

●著名资深一线测试专家●51testing高级讲师●高级咨询师  
首批ISTQB资格认证获得者 黄文高 倾情分享



# 深入性能测试

# LoadRunner

性能测试、流程、监控、调优全程实战剖析

黄文高 何月顺 编著

超过10年的软件测试经验，干货遍地

全程深入覆盖软件测试、测试流程、监控及调优

附赠价值3800元、6.97G、534分钟的高清测试实战视频DVD光盘

013064544

TP311.56  
1152

# 深入性能测试—— LoadRunner 性能测试、流程、 监控、调优全程实战剖析

黄文高 何月顺 编著



中国水利水电出版社  
www.waterpub.com.cn



北航 C1672001

TP311.56  
1152  
P

013084244

### 内 容 提 要

本书主要介绍如何使用 LoadRunner 进行性能测试工作，主要包括四大部分：入门篇、提高篇、监控篇和实战篇。入门篇主要讲述性能测试的基础知识，对 LoadRunner 进行简单介绍，让读者从整体上了解 LoadRunner。提高篇主要讲述性能测试工具 LoadRunner 三大组件的使用技巧，以及测试过程中需要注意的细节。监控篇主要讲述在整个性能测试过程中应该如何监控系统资源、Web 服务器和数据库，以及常用的调优技巧，并就性能测试的流程进行详细的介绍。实战篇通过两个案例：C/S 架构和 B/S 架构来介绍性能测试的整个过程。

本书适合性能测试工程师、资深测试工程师、测试经理、测试总监使用。

### 图书在版编目（C I P）数据

深入性能测试：LoadRunner性能测试、流程、监控、调优全程实战剖析 / 黄文高，何月顺编著. -- 北京：中国水利水电出版社，2013.6  
ISBN 978-7-5170-0994-8

I. ①深… II. ①黄… ②何… III. ①性能试验—软件工具 IV. ①TP311.56

中国版本图书馆CIP数据核字(2013)第145865号

策划编辑：周春元 责任编辑：张玉玲 加工编辑：刘晶平

书 名	深入性能测试——LoadRunner 性能测试、流程、监控、调优 全程实战剖析
作 者	黄文高 何月顺 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
经 售	电话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心 (零售) 电话: (010) 88383994、63202643、68545874 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	北京蓝空印刷厂
规 格	184mm×240mm 16开本 33印张 765千字
版 次	2013年6月第1版 2013年6月第1次印刷
印 数	0001—4000册
定 价	88.00元 (赠1DVD)

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换  
版权所有·侵权必究

# 前 言

## 12306，你懂的

每逢过年过节大家订票回家或出差旅游时，铁路 12306 订票网站几乎都会出现故障。很多人尤其是软件开发人员都在想一个问题：12306 订票网的性能怎么就这么差呢？不错，这是用户对这个网站的直观感受，这个性能表现的现象就是大家无法订票，而官方给出的系统每日的点击量超过 14 亿，这相当于全中国每个人都点击了一次，如果单纯从这个数据来看，似乎订不了票不是 12306 网站的错，而是订票人太多的缘故，但仔细分析一下会发现这样一个问题，虽然 12306 网站被频繁地点击，但是每当登录的人很多时都会出现这样的提示：“当前访问用户过多，请稍后重试！”，这就相当于门外有很多人敲门，但屋子里的人一直不开门一样，所以服务器根本就没有承受那么大的压力，又一次被忽悠了，其实市民的要求很简单，直接把每天从 12306 网站订出票的张数公布出来就可以，这可以直接反映出系统处理业务的能力，好理解又很简单，不用费脑子去思考“点击量”是什么意思。

从 12306 网站事件不难看出，在现在的软件质量体系中软件性能的重要性，而软件的性能必须依赖性能测试来验证，所以性能测试在未来软件测试体系中的地位显然是越来越重要，也越来越受企业重视的。

## 性能测试学习过程中的典型误区

在性能测试学习过程中最容易遇到以下两个典型误区：

### （1）学好 LoadRunner 就等于学好性能测试。

很多朋友认为性能测试主要是学习性能测试工具，其实并不是这么回事，性能测试工具只能说是性能测试的一个组成部分，并不能与性能测试等同，其实随着自身对性能测试的认识，你会发现性能测试工具更多的是用于模拟客户端产生压力的工具，其在性能分析和调优方面给出的数据支持相对来说较弱，所以仅仅靠性能测试工具是远远不够的，还需要使用其他一些监控和调优工具，才能做好性能测试。此外，性能测试计划也很重要，如果计划不当，那么测试出来的性能数据就不准确，所以性能测试不仅仅是工具，还有计划、监控和调优。

### （2）忽视性能测试过程。

对于一些有性能测试相关工作经验的性能测试工程师来说，很多人花很多时间去学习性能调优，当然这个并没有什么错，但是当调优的技能积累到一定程度后，又会发现自己在进行

性能测试时总是缺少了点什么，导致性能测试总是做得不理想，而这部分被“缺失”的内容就是性能测试过程或者说是性能测试流程，这也可能是很多读者比较容易忽视的一部分内容。性能测试过程是进行性能测试前的准备阶段，试想如果在进行性能测试之前没有一个好的性能测试方案来指导如何进行性能测试，那么就将导致测试出来的性能数据是错误的，而测试的数据都出现了错误，那么调优也就失去了意义。所以读者不应该忽视性能测试的过程，在学习性能测试过程中需要对性能测试的流程有一个很深刻的理解，这样才能帮助我们做出正确的测试方案，特别是业务模型和场景模型的定义，这是性能测试过程中的重中之重，并且只有对性能测试的流程有了相当程度的了解后，才能有序地梳理性能测试的过程，不至于让整个性能测试团队的工作处于混乱状态之中，才能更好地提高性能测试的效果。

## 关于本书

### (1) 本书解决读者哪些学习问题。

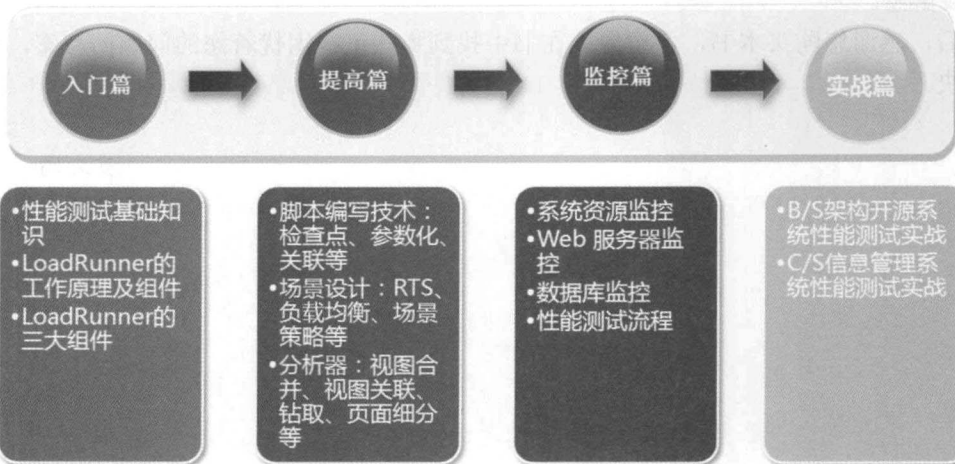
读者朋友买书都希望在书中学到一些可以使用的东西或笔者思考问题的方式，那么本书主要帮助读者解决哪些问题呢？

- 通过对本书的学习，可以熟练地使用性能测试工具 LoadRunner。书中详细介绍了 LoadRunner 的使用，特别突出了关键知识点（如关联、参数化等）的介绍，并且使用很多案例来介绍这些知识点的使用，这样可以更好地解决实际测试过程中的问题。
- 帮助读者提高监控和调优的技能，对一些有经验的朋友更希望看到该部分内容，而本书系统且全面地介绍了这方面的内容，并就其监控和调优的步骤进行了详细描述，这样可以更好地帮助读者掌握性能测试的技能。
- 熟悉性能测试流程，帮助读者更好地规范性能测试流程。笔者在与做性能测试的朋友交流时，发现很多朋友都会提及这样一个问题：对性能测试工具 LoadRunner 使用得很熟练，在性能测试过程中会进行监控和调优，但感觉还是有点乱，有点没有条理的感觉，其实很大的一个原因就是性能测试的流程不熟悉，导致总是没有一个规范的流程来指导如何进行性能测试。本书详细介绍了性能测试的流程，希望可以更好地帮助读者规范性能测试过程。

### (2) 本书的 4 个特点。

- 结构清晰，内容安排由浅入深，对初学者来说可以很轻松地入门，并且在描述概念的过程时尽量使用生活中的案例，便于读者对相关内容的理解。书中还详细描述了性能测试的流程、性能测试过程中如何监控与调优等，最大限度帮助那些有性能测试经验的读者朋友。本书包括四大部分：入门篇、提高篇、监控篇和实战篇，具体章节结构如下图所示。

# 深入性能测试



- 一些更关注于提高、调优方面的书籍并未对性能测试工具 LoadRunner 的使用进行详细描述，而本书详细描述了性能测试工具 LoadRunner 的使用，并就使用过程中需要注意的问题进行了详细讲解。
- 丰富了性能测试过程中监控和调优的内容。本书主要从系统资源、Web 服务器和数据库 3 个方面介绍了性能测试过程中的监控和调优技术，并将这几个方面的内容全面地展现出来，不仅仅是某个单方面的内容。
- 详细地介绍了性能测试的流程。在同类书中均未详细介绍性能测试的流程，但性能测试流程是规范性能测试、提高性能测试效率的一个重要环节，所以本书对这部分内容进行了详细阐述。

学习是一个漫长的过程，并且必须每天坚持，只有这样才能让自己不断地进步，而坚持是一个很痛苦的过程，所以有句话是这样说的：“成功必须要超越寂寞”。在学习过程中应该学会思考、善于总结，而不仅仅是看书，还要学会问为什么，笔者同样希望读者朋友在看此书的过程中将工作中的实践情况与本书中描述的内容相结合，将知识与工作经验更好地结合起来，这样才是我们真正需要的看书过程，希望读者朋友在本书中学到一些工作中需要的知识，祝读者朋友们学习愉快。

## 致谢

经过一年的努力，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和家人。

由于笔者水平有限，书中出现错误在所难免，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 [arivnhuang@163.com](mailto:arivnhuang@163.com)，作者将尽力给您答疑解惑。

最后，感谢您购买本书，希望您能在书中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  
2013年5月

# 目 录

前言

## 第一部分 入门篇

第 1 章 性能测试基础知识	2	2.2 LoadRunner 工作原理	16
1.1 软件性能概述	2	2.3 LoadRunner 工作过程	17
1.2 性能测试相关术语	4	2.4 LoadRunner 内部结构	18
1.2.1 响应时间	4	2.5 LoadRunner 11.0 特性	20
1.2.2 并发用户数	4	2.6 LoadRunner 性能测试步骤	21
1.2.3 吞吐量	5	2.7 小结	25
1.2.4 吞吐率	7	第 3 章 Vuser 发生器	26
1.2.5 TPS	7	3.1 脚本录制	26
1.2.6 点击率	7	3.1.1 如何选择协议	27
1.2.7 资源利用率	7	3.1.2 开始录制脚本	31
1.2.8 性能计数器	7	3.2 Recording Options 设置	34
1.2.9 思考时间	8	3.2.1 Recording 选项卡	34
1.3 性能测试划分	9	3.2.2 Advanced 选项卡	43
1.3.1 负载测试	9	3.2.3 Correlation 选项卡	44
1.3.2 压力测试	9	3.3 Run-Time Settings 设置	44
1.3.3 配置测试	10	3.3.1 Run Logic 选项卡	45
1.3.4 并发测试	10	3.3.2 Pacing 选项卡	46
1.3.5 可靠性测试	10	3.3.3 Think Time 选项卡	47
1.3.6 基准测试	10	3.3.4 Miscellaneous 选项卡	49
1.3.7 各类测试执行阶段	11	3.3.5 Log 选项卡	50
1.4 性能测试应用领域	11	3.4 脚本完善	52
1.4.1 能力验证	11	3.4.1 插入事务	52
1.4.2 规划能力	12	3.4.2 插入集合点	57
1.4.3 性能调优	12	3.4.3 插入注释	57
1.4.4 缺陷发现	12	3.5 小结	58
1.5 小结	13	第 4 章 Controller 控制器	59
第 2 章 LoadRunner 基础知识	14	4.1 场景类型介绍	59
2.1 LoadRunner 简介	14	4.1.1 手动测试场景	59



4.1.2 面向目标测试场景	61	5.1.3 Analysis 图	103
4.2 场景设计	62	5.2 摘要报告	105
4.2.1 手动场景 Schedule 配置	62	5.2.1 概要部分	105
4.2.2 面向目标场景 Schedule 配置	67	5.2.2 统计部分	106
4.2.3 配置 View Script	72	5.2.3 事务统计部分	106
4.2.4 配置 Load Generator	73	5.2.4 SLA	108
4.3 场景执行	76	5.2.5 HTTP 响应统计	115
4.3.1 场景控制	76	5.3 Analysis 常见图分析	118
4.3.2 场景执行期间查看场景	80	5.3.1 Vuser 图	118
4.4 场景监视	85	5.3.2 点击率图	119
4.4.1 关于联机监控	85	5.3.3 平均事务响应时间图	120
4.4.2 监控器与度量	89	5.3.4 吞吐量图	121
4.4.3 联机监视器	93	5.4 Analysis 报告	122
4.5 小结	96	5.4.1 HTML 报告	122
第 5 章 Analysis 分析器	97	5.4.2 SLA 报告	123
5.1 Analysis 简介	97	5.4.3 自定义报告	124
5.1.1 Analysis 基础知识	97	5.4.4 使用报告模板定义报告	126
5.1.2 设置选项	99	5.5 小结	126

## 第二部分 提高篇

第 6 章 脚本编写	128	6.4.2 录制中关联	167
6.1 检查点	128	6.4.3 录制后关联	170
6.1.1 插入检查点的原因	128	6.4.4 手动关联	172
6.1.2 插入检查点	132	6.4.5 关联函数介绍	176
6.1.3 检查点函数	139	6.4.6 关联技术的经典使用	182
6.1.4 通过检查点判断事务结束状态	141	6.4.7 关联与参数化的区别	185
6.2 Block (块) 技术	144	6.5 小结	186
6.3 参数化技术	148	第 7 章 场景设计实践	187
6.3.1 参数化的原因及条件	148	7.1 集合点	187
6.3.2 创建参数	149	7.1.1 集合点设置	187
6.3.3 参数类型属性	151	7.1.2 集合点与事务的关系	190
6.3.4 数据文件	155	7.2 IP 欺骗技术	192
6.3.5 导入数据	161	7.2.1 IP Spoofer 设置	192
6.4 关联技术	165	7.2.2 Controller 中启动 IP Spoofer	195
6.4.1 关联的原理	166	7.3 负载均衡技术	197

7.4	RTS 设置	200	8.3	页面细分	216
7.5	执行路径转换	200	8.3.1	页面细分原理	216
7.5.1	路径转换介绍	200	8.3.2	实例讲解	219
7.5.2	编辑路径转换表	201	8.4	钻取技术	221
7.6	在 LoadRunner 中使用功能测试脚本	203	8.4.1	钻取技术原理	221
7.6.1	QuickTest 创建 GUI Vuser 脚本	204	8.4.2	实例讲解	222
7.6.2	WinRunner 创建 GUI Vuser 脚本	205	8.5	导入外部数据	223
7.6.3	场景中 使用 GUI Vuser 脚本	206	8.5.1	导入数据工具	223
7.7	小结	207	8.5.2	自定义文件格式	226
第 8 章	结果分析实践	208	8.6	使用 HTTPWatch 分析响应时间	228
8.1	分析图合并	208	8.6.1	HTTP 消息格式	228
8.1.1	分析图合并原理	208	8.6.2	Request Grid	230
8.1.2	实例讲解	210	8.6.3	Request View	231
8.2	分析图关联	212	8.6.4	Summary View	238
8.2.1	分析图关联原理	212	8.6.5	其他功能	239
8.2.2	实例讲解	214	8.7	小结	240

### 第三部分 监控篇

第 9 章	操作系统性能监控与调优	242	10.2	Apache 调优	291
9.1	Windows 操作系统监控	242	10.2.1	硬件与操作系统方面	291
9.1.1	LoadRunner 直接监控	242	10.2.2	运行时的配置	291
9.1.2	Windows 性能工具监控	243	10.2.3	编译时的配置	296
9.1.3	Windows 计数器	249	10.3	Tomcat 监控	301
9.2	Linux/UNIX 操作系统监控	255	10.3.1	Status 页监控	302
9.2.1	CPU 监控	255	10.3.2	JConsole 监控	304
9.2.2	内存监控	260	10.4	Tomcat 调优	316
9.2.3	磁盘监控	264	10.4.1	JVM 调优	316
9.2.4	网络监控	269	10.4.2	Tomcat 配置	323
9.3	nmon 系统资源监控工具	273	10.4.3	连接器配置	326
9.3.1	nmon 工作流程	273	10.4.4	APR 配置	332
9.3.2	nmon 命令	274	10.5	小结	336
9.3.3	结果分析	283	第 11 章	MS SQL 数据库监控与调优	337
9.4	小结	284	11.1	监控 SQL Server 资源	337
第 10 章	Web 服务器监控与调优	285	11.1.1	瓶颈类型	337
10.1	Apache 监控	285	11.1.2	内存瓶颈	338

11.1.3	CPU 瓶颈	340
11.1.4	磁盘瓶颈	341
11.1.5	Wait Statistics 监控	341
11.2	SQL Server 等待类型	342
11.2.1	SQL 等待类型	342
11.2.2	跟踪等待	352
11.2.3	阻塞与锁	354
11.3	SQL Profiler 监控查询	359
11.3.1	SQL Trace 相关术语	359
11.3.2	SQL Trace 选项	363
11.3.3	捕获阻塞事件	367
11.3.4	捕获 Showplan XML 数据	369
11.3.5	捕获死锁图	372
11.3.6	SQL Profiler 识别长时间查询	375
11.3.7	Profile Trace 与 System Monitor 关联	380
11.4	索引调优	382
11.4.1	索引原理	382
11.4.2	填充因子	383
11.4.3	聚集索引	384
11.4.4	非聚集索引	386
11.4.5	堆表	388
11.4.6	用 DTA 调校索引	389
11.4.7	索引维护	395
11.5	T-SQL 调优	399
11.5.1	NOT IN 和 NOT EXISTS	399
11.5.2	谓词的使用	399
11.5.3	为中间结果使用临时表	401
11.6	小结	402

第 12 章	特殊协议	403
12.1	Windows Sockets (WinSock) 协议	403
12.1.1	Windows Sockets 录制选项设置	404
12.1.2	Windows Sockets 录制	405
12.1.3	Windows Sockets 数据操作	409
12.1.4	关于 LRS 函数	413
12.2	邮件服务协议	418
12.2.1	邮件服务协议简介	418
12.2.2	邮件服务协议录制	420
12.2.3	脚本分析	426
12.2.4	关于 SMTP 和 POP3 函数	428
12.3	小结	430
第 13 章	性能测试过程	431
13.1	性能测试过程概述	431
13.2	性能测试设计	432
13.2.1	需求调研	432
13.2.2	业务模型	433
13.2.3	场景模型	435
13.2.4	数据设计	437
13.2.5	环境设计	438
13.3	性能测试构建	439
13.3.1	脚本开发	440
13.3.2	场景设计	441
13.3.3	搭建测试环境	441
13.3.4	准备数据	441
13.4	性能测试过程执行	442
13.5	性能测试分析、诊断、调节	442
13.6	小结	443

## 第四部分 实战篇

第 14 章	客户关系管理系统性能测试	445
14.1	系统介绍	445
14.2	需求分析	447
14.2.1	性能指标	447
14.2.2	需求详细分析	448
14.3	测试方案及计划	449
14.3.1	人力资源	449
14.3.2	时间进度	449

14.3.3	测试环境准备	450	15.3	测试方案及计划	481
14.3.4	业务模型创建	451	15.3.1	人力资源	481
14.3.5	场景模型创建	452	15.3.2	时间进度	481
14.3.6	测试数据准备	453	15.3.3	测试环境准备	482
14.4	测试用例	453	15.3.4	业务模型创建	483
14.5	执行测试	456	15.3.5	场景模型创建	483
14.5.1	脚本开发	456	15.3.6	测试数据准备	484
14.5.2	场景设计	461	15.4	测试用例	485
14.5.3	计数器设置	466	15.5	执行测试	486
14.5.4	场景监视	468	15.5.1	脚本开发	486
14.6	结果分析	470	15.5.2	场景设计	490
14.7	测试结论	478	15.5.3	计数器设置	496
14.8	小结	478	15.5.4	场景监控	496
<b>第 15 章</b>	<b>信息系统性能测试</b>	<b>479</b>	15.6	结果分析	498
15.1	系统介绍	479	15.7	测试结论	506
15.2	需求分析	480	15.8	小结	506
15.2.1	性能指标	480	附录 A	主要计数器	507
15.2.2	需求详细分析	480	附录 B	性能测试 i 模型	511

# 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介绍。

# 1

## 性能测试基础知识

在学习性能测试之前,有必要先了解一些性能测试的理论基础知识,为后期的性能测试做准备。需要了解什么是软件性能,性能测试需要关注的内容;了解在性能测试过程中常用的相关术语以及性能测试过程中需要关注的指标;了解性能测试的划分和应用领域,这样可以更好地确定需要进行哪些性能测试。

本章主要包括以下内容:

- 软件性能概述
- 性能测试相关术语
- 性能测试划分
- 性能测试应用领域

### 1.1 软件性能概述

什么是软件性能?大多数读者朋友会认为性能是一种指标,是反映软件系统或产品处理用户请求的响应时间,在软件质量模型中,性能被定义为软件的一种特性,软件质量模型如图 1-1 所示。

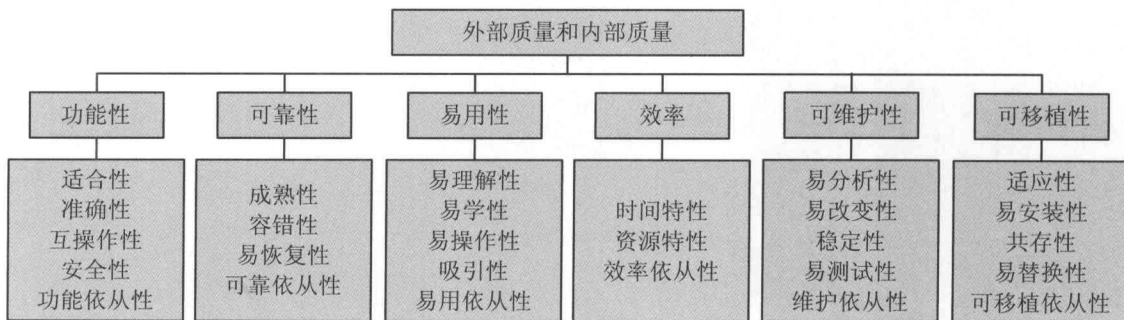


图 1-1 软件质量模型

在软件质量模型中效率特性即为软件的性能，其包含两个方面的特性：时间特性和资源特性。时间特性是指系统处理客户请求的响应时间；资源特性是指在进行性能测试过程中，系统资源消耗的情况，常见的系统资源主要包括处理器（CPU）、内存和磁盘的使用情况。所以通常说的软件性能不仅仅包括响应时间，还包括系统资源消耗。

虽然软件性能包含两个方面，但是不同的人所关注的性能层面有所不同（如用户只关注时间特性）。通常情况下，关注软件性能的人主要包括 3 类：用户、系统管理员与性能测试工程师、软件开发工程师。

### 1. 用户

从用户的角度来说，软件性能是软件系统对用户提交请求所响应的的时间。通俗地讲，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 1-2 所示。

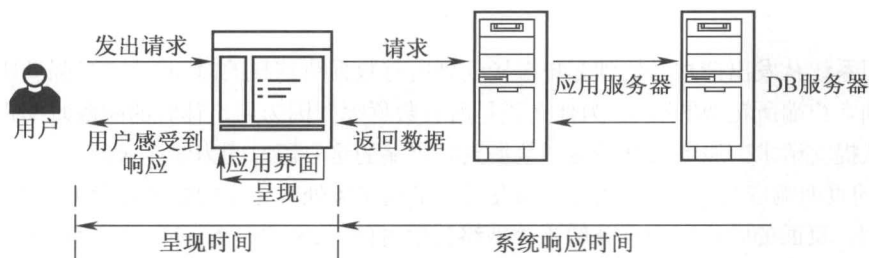


图 1-2 Web 系统响应

需要注意的是，用户体会到的“响应时间”，既有客观的成分，也有主观的成分。用户认为从提交业务到系统开始返回信息的时间为系统的响应时间。例如，用户执行某个操作，该操作会返回大量的数据，假如待返回的数据并不能同时返回到主界面，而是先将一部分数据呈现出来，再慢慢地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已经有一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。

### 2. 系统管理员与性能测试工程师

从系统管理员和性能测试工程师的角度来说，在响应时间方面的理解与用户完全一致。但系统管理员和性能测试工程师是一群特殊的用户群体，其不仅仅关注系统的响应时间，还关注服务器系统资源的使用情况。系统管理员及性能测试工程师之所以关注资源的消耗情况，是因为系统响应时间达到要求并不代表系统就能正确地处理客户端提交的请求，例如银行系统，假设在处理存款业务时，每笔业务的响应时间为 100ms，但当前的 CPU 和内存的使用率都已达到 90%，超过正常使用的阈值，这样虽然响应时间达到要求，但是不能保证服务器不出问题，因为服务器已经处于一个临界状态，很可能出现存款不成功的情况，如果这样的话，即使响应时间更短也不能达到性能的要求。

另外，如果测试系统配置时，管理员和性能测试工程师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 个用户并发没有问题，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、中间件及应用服务器等问题。

## 1.2 性能测试相关术语

了解了什么叫系统性能之后，需要对性能测试过程中常用的术语有一个详细的了解，为后面的性能测试做准备。性能测试过程中的常用术语有响应时间、并发用户数、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一段时间为响应时间。

以一个 Web 应用的页面响应时间为例，从客户端发送请求到服务处理完成的整个过程如图 1-3 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”（ $N_1+N_2+N_3+N_4$ ）和“应用服务器处理的时间”（ $A_1+A_3$ ），“数据库处理的时间”为  $A_2$ ，所以整个 Web 页面请求的响应时间为  $N_1+N_2+N_3+N_4+A_1+A_2+A_3$ 。

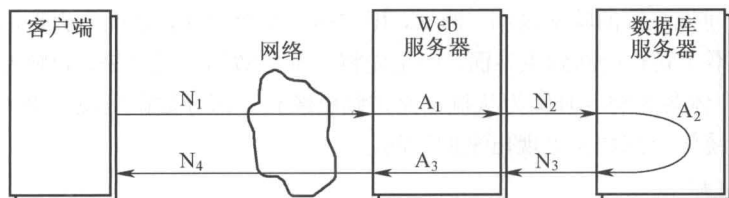


图 1-3 Web 页面响应时间分解

### 1.2.2 并发用户数

并发用户数是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了压力。

因此对于并发用户这个概念的理解经常会出现以下两个误区：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。OA（办公自动化）系统的并发用户数一般是在线用户的 5%~20%，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。

一般情况下，可以参考以下方法来确定性能测试时的并发用户数：

(1) 参考其他同类产品。

如果不知道测试过程中需要测试多少并发用户数，那么可以分析市场上同类产品测试的情况，参考其测试的并发用户。

(2) 分析历史数据。

如果有历史数据，可以分析后台统计到的历史数据，分析一年或半年的交易量，可以得到服务器每天需要处理的业务数量，进而可以确定系统需要支持的并发用户数。

(3) 试上线运行。

如果没有同类产品可以参考，那么试上线运行也是一种方法，如火车票官方订票系统，这类系统就没有可参考的对象，这样可以通过试上线运行的方法来大体了解终端用户提交业务的情况，进而确定系统每秒钟需要处理多少笔业务或 HTTP 请求数。

### 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的字节数，吞吐量的单位为 B/s，吞吐量的大小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，可以采用下面的公式来计算：

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

式中，F 为吞吐量， $N_{VU}$  为 VU（Virtual User，虚拟用户）的个数，R 为在 T 时间内每个 VU 发出的请求字节数，T 为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 1-4 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。