财政平台一体化应用系统性能测试①

张世琼

(安徽省财政信息中心，合肥 230061)

摘 要：性能测试是通过模拟真实环境下的负载，采集系统各方面的性能指标，评估系统正常运行情况下的承受力和稳定性，分析系统的性能与存在的瓶颈。应用 LoadRunner 测试工具，从性能指标和性能测试方法两方面，深入地分析了性能测试过程和方法。根据县级财政平台一体化应用系统特点，设计模拟运行真实的业务场景，采用压力测试和负载测试获取性能指标，运用网页元素细分和关联度等进行详细分析性能瓶颈，并提出改善方法。

关键词：性能瓶颈；软件性能测试；性能指标；LoadRunner 工具；并发

Performance Testing of Finance Integration Application Platform

ZHANG Shi-Qiong

(Anhui Finance Information Center, Hefei 230061, China)

Abstract: The Performance testing is the process of collecting performance indexes of all system aspects through simulating load in real conditions to evaluate system endurance and stability under normal operating conditions, and to analyze system performance and bottlenecks. Using LoadRunner testing tools, this paper introduced detailed process and methods of performance test, through performance index and performance testing methods. According to the County Level Finance Integration Application Platform features, this article designed simulated real business scenarios, obtained performance index using pressure testing and load testing, analyzed performance bottleneck through web page diagnostics and graphs for correlation, and brought improving methods.

Key words: performance bottleneck; performance testing; performance criteria; LoadRunner tool; concurrency

采用性能测试是保证软件质量的重要手段，性能测试通过负载测试、压力测试等方法，监控系统资源，不仅验证应用系统所具备的能力指标，而且针对系统性能如系统反应速度慢或者服务中断，甚至宕机等薄弱环节，测试应用系统能否承受大量的并发用户数以及快速响应用户发送的请求，能否长时间稳定运行，获取性能指标并进行分析找出系统性能瓶颈，使整个系统的性能得到最大的优化，从而验证系统能力和不断改善系统性能[1]。本文结合安徽省县级财政一体化应用系统测试，深入分析性能指标和性能测试方法，设计模拟运行真实的业务场景，采用 LoadRunner 测试工具[2]，使用压力测试和负载测试获取性能指标，运用网页元素细分和关联度等进行详细分析性能瓶颈，并提出改善方法。

1 性能测试方法

性能测试是通过模拟真实环境下的负载，采集系统各方面的性能数据，评估系统正常运行情况下的承受力和稳定性，分析系统的性能与存在的瓶颈。测试方法主要包括负载测试（Load Testing）、压力测试(Stress Testing)、配置测试（Configuration Testing）并发测试（Concurrency Testing）和可靠性测试（Reliability Testing）等[3]。在测试过程中，通常需要合理的结合如下几种测试方法，模拟真实环境，设计不同的测试

① 收稿时间:2011-10-18;收到修改稿时间:2011-11-25

场景获取更多有效的性能指标。

1.1 负载测试 Load Testing

负载测试是为了找到系统最大的负载能力，通过对被测试系统不断加压，直到超过预定的指标或者系统部分资源已经达到饱和状态不能再加压为止。为确保负载测试的有效性。通常借助自动化测试工具，尽量模拟真实用户的操作，执行测试。由于时间和资源有限，负载测试不可能对应用系统所有功能操作进行性能测试，因此，负载测试期间需选择几项主要功能进行测试，所选择测试策略包括独立业务性能测试和混合业务性能测试。

1.2 压力测试 Stress Testing

压力测试是指测试系统已经达到饱和程度时，系统处理业务能力。一般来说先进行系统的负载测试，确认出各种性能指标的极限数据后，再执行压力测试。对系统不断的持续加压，系统提供服务出错或崩溃，可暴露出系统存在的瓶颈。判断服务器资源是否成为瓶颈，重点应关注系统资源性能指标、事务成功数、事务失败数、事务响应时间等性能指标。

1.3 配置测试 Configuration Testing

配置测试是通过系统环境设置调整系统软硬件进行测试，了解和分析各种不同环境对系统性能的影响，从而找到系统各项资源的最优分配原则。配置测试主要用于性能调优，在经过测试获得了基准测试数据后进行环境调整（包括硬件资源，网络，应用服务器等），再将测试结果与基准测试数据进行对比，判断是否达到最佳状态。

1.4 并发测试 Concurrency Testing

并发测试是通过模拟用户并发访问，测试多用户访问同一应用、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性能问题。大量并发用户模拟和相关数据的准备借助自动化测试工具来生成。

1.5 可靠性测试 Reliability Testing

可靠性测试是指测试系统在一定的业务压力下，系统可持续运行的时间。此测试是考察在规定的测试时间内系统性能指标，例如持续无故障运行天数等，观察是否达到系统要求的稳定性。

2 性能指标概述

如何评价一个应用系统的性能及寻找系统瓶颈？性能测试的目的一般分为以下两部分：一部分是查看系统在一定的负载条件下运行是否合乎需求；另一部分是通过系统在一定负载下运行所得的性能指标数据，分析它可能发生性能瓶颈的地方。因此，能够对当前的系统产生一定的负载，并且能够分析系统在一定负载下的表现是实现性能测试的关键。为有效地利用自动化的测试工具模拟多种正常、峰值以及异常负载条件来测试系统的各项性能指标，分析测试结果，需要筛选分析影响系统的主要测试性能因素[4]，在筛选出测试指标中就应该给出本次性能测试需要关注的具体项，做到有的放矢。如图 1 所示应用系统测试性能因素。

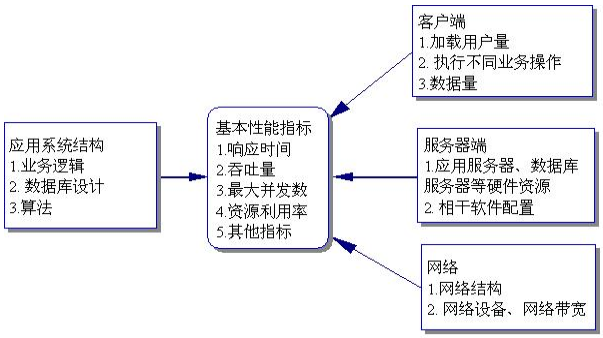

图 1 应用系统测试性能因素

系统性能主要通过性能指标体现，进而通过分析性能数据寻找系统的瓶颈。因此，性能指标的筛选在性能测试过程中非常重要。应用系统性能测试主要包括以下几类基本性能指标：

2.1 响应时间

响应时间是用户发送请求到客户端接收服务器返回的结果所经历的时间。通过观察响应时间快速增长时其他性能数据的变化。寻找系统可能存在的瓶颈。例如大量并发用户同时读写数据库，服务器面临的负担较重．响应时间就会增加；当系统的某一种或几种资源耗尽时，响应时间急剧增加。响应时间涉及的具体性能指标包括最大事务响应时间、平均事务响应时间、最小事务响应时间。

2.2 吞吐量、点击率、每秒事务数(TPS)

吞吐量是单位时间内系统所处理的用户请求数。一般是用系统每秒字节数或请求数来衡量。该指标反映了系统的事务处理能力，通常用户请求数越大．吞吐量就越大，当用户请求数达到一定值，吞吐量逐渐处于饱和

状态并可能出现拐点，此时随着负载的不断增加，系统的性能会逐渐降低。可以依据服务器的吞吐量来评估虚拟用户产生的负载量，观察服务器在流量方面的处理能力以及是否存在瓶颈。点击率是每秒服务器处理的 HTTP 申请数。通过它可以评估虚拟用户产生的负载量，如将其和“平均事务响应时间”图比较，可以查看点击次数对事务性能产生的影响。“每秒通过事务数/TPS”显示在场景运行的每一秒钟，每个事务通过、失败以及停止的数量，是考察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时间事务负载。分析 TPS 主要是看曲线的性能走向。当增大系统的压力例如增加并发用户数时，吞吐率和 TPS 的变化曲线呈大体一致，则系统基本稳定；若压力增大时，吞吐率的曲线增加到一定程度后出现变化缓慢，甚至平坦，很可能是网络出现带宽瓶颈，同理若点击率和 TPS 曲线出现变化缓慢或者平坦，说明服务器开始出现瓶颈[5]。

2.3 资源利用率

资源利用率是系统运行过程中各种资源的使用情况，一般包括硬件、操作系统、网络、数据库等方面，例如服务器的 CPU、内存、磁盘、网络带宽等。通常系统资源利用率与加载用户数成正比，当资源利用率保持或长期接近 90%，说明该资源已经成为系统的瓶颈，通过提升资源的容量可缩短响应时间，改善系统性能。

3 测试过程与结果分析

县级财政一体化应用系统包括预算指标管理、国库集中支付、工资统发、报表管理等功能，是全面覆盖财政预算及收支管理的金财应用工程。县级财政一体化应用系统体系由客户端浏览器 IE、Web 应用服务器、Weblogic 中间件应用服务器和 Oracle 数据库服务器等构成的 B/S 三层结构，如图 2 所示。

针对县财财政平台一体化应用系统在试点推广运行阶段中，存在业务系统响应时间过长和系统运行中的不稳定等问题，需要对县财系统平台一体化应用系统进行性能测试。计划模拟运行真实的业务场景，采用压力测试和负载测试获取性能指标，分析测试指标结果找出性能瓶颈，从而改善应用系统，为我省财政平台一体化应用系统的全面推广起到指导性作用。

3.1 测试环境

测试工具选用是 HP LoadRunner11.0，LR 是一种预测系统行为和性能的负载测试工具，以模拟成千上万用户并发负载实时监测系统性能的方式来确认和查找问题，能够对整个企业架构进行测试[6]。通过使用 LR 测试工具，能最大限度地缩短测试时间，优化性能。LR 主要分成三个部分：测试脚本生成工具 Virtual User Generator、测试场景运行及监视工具 Controller、测试结果分析工具 Analysis。县级财政一体化应用系统的测试环境如表 1 所示。

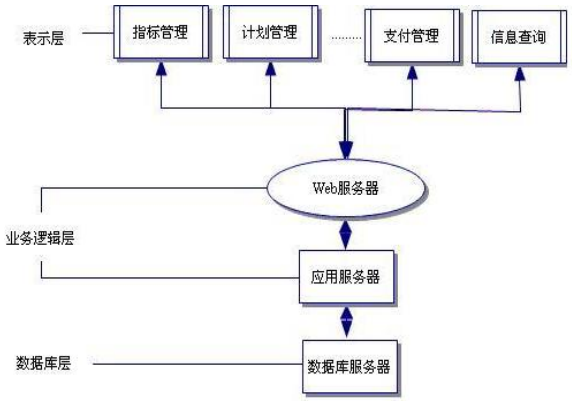

图 2　县级财政一体化应用系统结构图

表 1　县级财政一体化应用系统测试环境表

序号	设备名称	操作系统/应用软件	配置信息
1	数据库服务器	Windows Svr 2008 R2 64/Oracle 10g	HP Proliant DL580 G7 32G 内存 6*300GB 硬盘
2	应用服务器	Weblogic 8.1 SP4	
3	安徽财政一体化应用系统	县区版财政一体化管理信息系统 1.0	

3.2 基础数据

县级财政一体化应用系统的运行数据库是按年结转的，因此基础数据设计的标准是以试点县长丰县财政局试运行的 1-10 月份真实数据量为参照而制定，如表 2 所示。

表 2　基础测试数据量

序号	业务数据	数据量
1	指标	100
2	计划	1597
4	直接支付	5222
5	授权支付	399
6	银行确认支付	5505

3.3 并发用户数

对于平台一体化应用大型系统，都有业务集中使用的情况出现，即系统使用的高峰。使用高峰可能出现在一天、一月、一年中的某个时间点上或时间段上。例如在同一天内，大多数系统的使用情况都会随着时间发生变化，对于不同的业务高峰对应的时间也不同，如表 3 所示。确定系统使用高峰时段，有利于我们进一步确定系统在负载测试时，确定并发用户数的需求。

表 3 长丰县财政一体化应用系统用户蜂值表

用户分类	08:00-08:30	08:30-09:00	09:00-09:30	09:30-10:00	10:00-10:30	10:30-11:00	11:00-11:30	11:30-12:00	14:00-14:30	14:30-15:00	15:00-15:30	15:30-16:00	16:00-16:30	16:30-17:00	17:00-17:30	17:30-18:00
代理银行		2		1						1						1
预算单位	32	37	37	39	27	40	26	14	25	26	33	55	66	42	28	9
业务处室	10	16	18	14	16	26	20	10	2	16	18	12	18	11	8	8
用户蜂值	39	50	50	51	30	51	31	21	27	42	38	65	69	45	32	13
平均用户	11	18	18	19	15	19	16	7	6	12	19	17	17	15	14	5
合计用户	42	55	55	54	43	66	46	24	27	43	51	67	84	53	36	18

在实际的性能测试中，一般比较关注业务的并发用户数的测算，公式(1)和(2)中用来估算并发用户数和峰值[7]。其中 C 是平均的并发用户数，Cp 是并发用户数峰值，Login session 定义为用户登录进入系统到退出系统的时间段，n 是 Login session 的数量，L 是 Login session 的平均长度，T 是考察的时间长度，假设用户的 Login session 产生符合泊松分布，则公式并发用户数和峰值为：

$$C=n*L / T \quad (1)$$

$$Cp=C+3*\sqrt{C} \quad (2)$$

经实际统计测算县级财政一体化应用系统平均用户数是 100 个，用户在 8 小时工作时间内使用系统，一个典型用户，一天内从登录到退出系统的平均时间为 4 小时。据公式(1)和(2)可计算平均并发用户数和峰值用户数为：

C=100*4/8=50，

Cp=50+3*$\sqrt{50}$ =71。

这与县级财政一体化应用系统试点运行用户蜂值统计结果（表三所示）的实际需求分析对照基本吻合。

3.4 测试过程

我们认为县区财政作为一体化应用系统，最常使用并且对于系统的整体性能有着较大影响的是“指标管理”、“计划管理”、“支付管理”、“银行管理”和“综合信息查询”五个应用系统模块。每个系统业务相互关联，前个系统业务处理数据是后序系统业务处理的数据源。针对该系统的特点，从中挑选出具有代表性的功能点，作为本次性能测试的用例。在设计测试场景时，不仅严格按照业务数据的处理关联和顺序，并且为真实模拟实际业务，需要设计多业务混合场景。

3.4.1 单个业务场景设计

单个业务场景设计包括计划管理、支付管理等场景。例如在计划管理中“录入保存并继续”场景中，设置集合点分别并发 40 个用户、60 个用户和 80 个用户，在同一时刻并发做一般计划的录入保存并继续操作。其事务响应时间如表 4 所示。

表 4 “录入保存并继续”事务响应时间表

并发用户数	最小值	平均值	最大值	成功事务数（%）
40	0.488s	3.67s	7.244s	774 （100%）
60	0.593s	5.987s	10.755s	967 （100%）
80	0.693s	6.281s	11.103s	1253 （100%）

3.4.2 多业务混合场景设计

根据实际业务来考虑模拟真实业务环境，在多业务混合场景中针对不同并发用户人数进行不同业务操作，包括指标执行情况查询、计划管理、支付管理系统各个重点功能，一般计划录入 30 个用户和审核 1 个用户（平均每次审核单据量 60 条），授权录入 10 个用户、直接支付录入 30 个用户、指标执行情况查询 20 个用户。其混合场景中，每 5s 增加 2 个用户，100 个用户全部登陆系统做完各自操作后退出系统，主要测试的是在多个场景客户同时进行多事务操作时系统响应情况。测试响应时间结果如表 5 所示。

表 5 多业务混合场景事务响应时间表

事务	业务	最小值	最大值	平均值
1	授权支付保存并继续	0.21s	3.341s	1.6s
2	授权支付保存并退出	0.735s	4.825s	3.102s
3	授权支付查询	0.739s	15.568s	4.837s
4	一般计划保存并退出	0.742s	5.601s	3.811s
5	一般计划审核	1.244s	4.31s	3.495s
6	一般计划新增查询	0.118s	9.876s	6.786s
7	直接支付保存并继续	0.546s	9.471s	4.152s
8	直接支付保存并退出	0.26s	8.717s	3.199s
9	直接支付新增查询	0.074s	7.983s	2.439s
10	指标执行情况查询	0.108s	2.157s	0.922s

多业务混合场景下，对应各个平均事务响应时间和虚拟用户合并图，如图 3 所示。

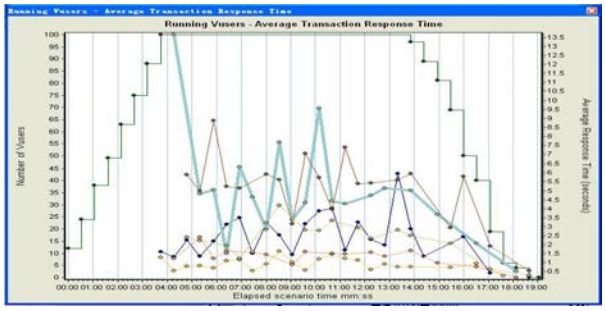

图 3　混合场景平均事务响应时间和虚拟用户合并图

3.5 测试分析

通过单个业务场景测试结果可以看出,“一般计划保存并继续”随着并发用户增加至 80 个，平均响应时间增加明显，需要增大压力测试；此外，通过多业务混合场景分析，在多个日常业务操作时各个功能响应情况，绝大部分功能平均响应时间在 5s 内；系统资源如表 6 所示，观察发现 CPU 利用率及可用内存等系统资源相关指标适中，排除多业务混合场景下系统资源的瓶颈。

表 6　混合场景下 CPU 利用率及可用内存页面交换率

服务器	CPU 利用率（%）			页面交换率（Page/sec）		
	最小	最大	平均值	最小	最大	平均值
应用服务器	0	39.18	10.14	0	1513.93	252.74
数据库服务器	0	74.73	26.93	0	0.36	0.02

对“一般计划保存并继续”事务增大压力测试，发现系统的最大承受负载压力为 100 个并发虚拟用户。此时，系统的处理速度明显下降，当继续增加至 150 个并发虚拟用户加载测试时，事务成功率下降为 96.1%,显然性能存在瓶颈。如图 4 所示，“一般计划保存并继续”事务测试加载 100 并发用户数和响应时间的合并图。

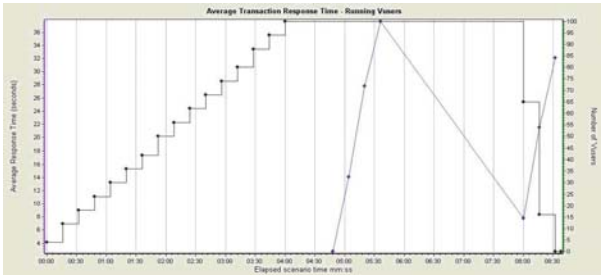

图 4　100 并发用户和响应时间合并图

图 4 显示，该事务随着加载并发用户数逐渐增加，其响应时间急剧上升达到 30s，响应时间超过用户可接受的最大闸值。同时，CPU 利用率及可用内存等系统资源相关指标如图 5 所示。

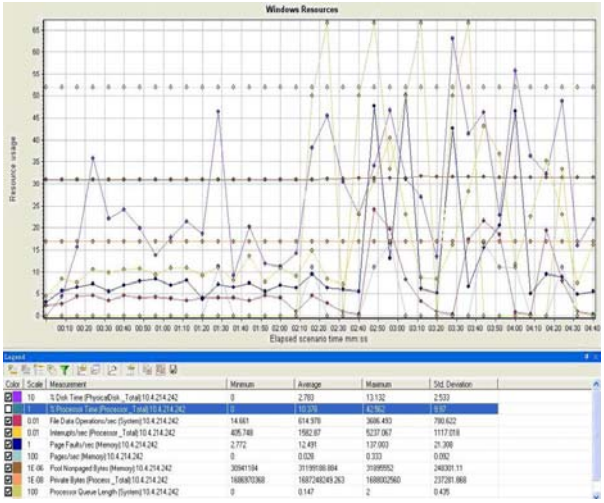

图 5　CPU 利用率及可用内存等系统资源相关指标

运用 LoadRunner“网页元素细分”分析,来评估页面内容是否影响事务的响应时间，通过它可以深入地分析页面中那些元素组件响应时间较大。网页元素细分结果如图 6 所示。

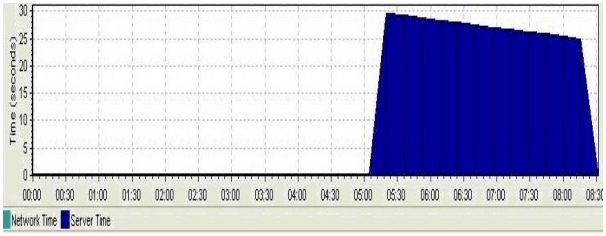

图 6　应用服务器响应时间与网络响应时间对比

应用服务器处理时间过长 Server Time>28s，网络 Network Time<0.1s，网络传输质量较好。因此，该响应时间是由应用服务器性能产生的。网络传输时间可以忽略不计。可考虑从应用层面优化业务逻辑，通过改进程序算法等方式缩短应用服务器处理时间。

进一步应用 LoadRunner 关联图分析结果如图 7 显示，应用服务器 CPU 利用率与页面错误 Page Faults/sec 指标高度关联达 70%，存在系统颠簸可能造成过量调页而消耗资源，影响系统响应时间性能指标。

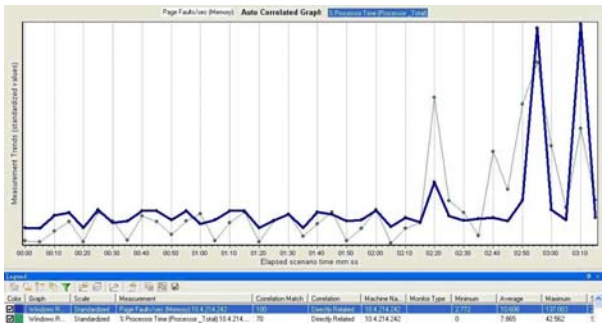

图 7 CPU 利用率与页面错误 Page Faults/sec 关联图

综上分析，发现“一般计划保存并继续”事务是系统性能瓶颈所在，需进一步优化修改。经程序代码排查，该段程序设计中，需要查询相关项目编号等信息，消耗较多的数据库资源。解决办法可考虑对 weblogic 应用服务器数据库连接池的配置优化，增大数据库的最大连接数。同时，从数据库设计和优化搜索算法两方面考虑修改此段程序，从而缩短响应时间，提高系统性能。

4 性能测试方法

在性能测试中，需要针对应用系统业务特点，模拟运行真实的业务场景，灵活运用 LoadRunner 性能测试工具，从性能指标和性能测试方法两方面分析入手，合理选择压力测试和负载测试等多种测试方法，运用多角度网页元素细分和关联度等分析工具进行详细分析测试性能指标结果，找出了该系统的性能瓶颈。性能测试中的测试分析是难点，在应用测试实践中，需要针对具体测试结果，调整测试分析思路，进而改进测试方法和步骤。同时，针对性能瓶颈，应该采用多种有效方法优化应用系统服务器参数配置和应用系统的程序代码，从而使整个系统性能提高，达到性能测试目的。

参考文献

1 陈霁,牛霜霞,龚永鑫.性能测试进阶指南.北京:电子工业出版社,2009.14–18.

2 桑圣洪,胡飞.性能测试工具LoadRunner的工作机理及关键技术研究.科学技术与工程,2007,(6):1019–1022.

3 黄文高,赵丹.LoadRunner 性能测试完全讲义.北京:中国水利邮电出版社,2010.6–8.

4 李艳芹,陈跃华,郭松柏.基于 Web 应用系统的性能测试综述.电脑知识与技术,2010,6(28):8014–8017.

5 董跃华,彭稷栋.利用 LoadRunner 实现网页负载压力测试.江西理工大学学报,2010,31(5):52–56.

6 Hewlett-Packard Development Company. What’s New in HP performance Center 11.00&HP LoadRunner 11.0,2011. http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA1-2118ENW.pdf.

7 段念.软件性能测试过程祥解与案例剖析.北京:清华大学出版社,2006.10–12.

(上接第 12 页)

参考文献

1 Chen Ji H. Language specific issue and feature exploration in Chinese event extraction. Proc. of NAACL HLT 2009. Boulder, Colorado. 2009.209–212.

2 Metwally A, Agrawal D, Abbadi AE. Efficient computation of frequent and top-k elements in data streams. Proc. of the 10th International Conference on Database Theory (ICDT 2005). LNCS 3363, Berlin: Springer-Verlag. 2005. 398–412.

3 任仲晟,薛永生.基于页面标签的 Web 结构化数据抽取.计算机科学,2007,10:133–136.

4 刘书华,陈国奎.基于 PowerBuilder 的网页数据抓取.计算机系统应用,2009,18(2):171–175.

5 秦进,陈笑蓉.文本分类中的特征抽取,计算机应用,2003,23(2):45–46.

6 许建潮,胡明.中文 Web 文本的特征获取与分类.计算机工程,2005,31(8):24–25,39.