

基于用户会话的 Web 应用性能测试方法的研究

梁力图 陆 璐

(华南理工大学计算机科学与工程学院 广州 510006)

摘 要 近年来,Web 应用成为互联网时代基础设施中最重要的组成部分。随着互联网技术的不断进步以及使用用户数量的不断增长,Web 应用也越发变得复杂。为了保证 Web 应用性能测试的有效性和真实性,提出一种利用服务器日志中的用户会话信息、结合经过序列匹配算法改进的层次凝聚算法来自动产生 Web 应用性能测试脚本的方法。提出的方法通过从服务器日志中提取出用户会话信息,分析用户真实的访问模式,通过聚类算法将同类型的用户访问模式进行聚集,自动产生相应的测试请求序列,生成测试集。相比于传统的基于录制/回放的测试方法,基于用户会话的性能测试方法通过对用户会话进行聚类分析,还原实际的用户访问模式,减少了设计测试用例时的人工参与,保证了测试结果的有效性和真实性。

关键词 用户会话,层次聚类,性能测试,有效性

中图法分类号 TP331

文献标识码 A

DOI 10.11896/j.issn.1002-137X.2014.11.010

Modified Hierarchy Clustering Algorithm for User-session-based Performance Testing

LIANG Li-tu LU Lu

(School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract Web applications are important parts of global information infrastructures. It attracts more and more researchers to study Web application performance testing. In this paper, a user-session-based approach combined with a modified agglomerative hierarchical clustering algorithm was proposed to automatically generate performance test cases. The approach generates test cases by exploiting user-sessions from server logs. It can not only reduce the manual effort when test engineers design the test cases, but also guarantee the validity of the testing results. In our approach, we firstly gave a definition of how to achieve the similarities between two URLs, and then employed a dynamic programming algorithm to calculate the similarities between two user sessions. According to the similarity matrix, a bottom-up agglomerative hierarchical clustering was employed to cluster the user sessions and then generated the test cases. Finally, experimental result of our approach shows its validity.

Keywords User session, Agglomerative hierarchical clustering, Performance testing, Validity

1 引言

近年来,Web 应用成为全球信息基础设施中最重要的组成部分,为全世界各类用户提供各种各样的服务。随着技术的进步和用户需求的不断变化,Web 应用也变得越来越复杂,它们提供网络存储、网络计算、高质量视频点播、电子商务等,一个大型的 Web 应用有着数十万乃至百万的用户群。因此,一个 Web 应用的性能和性能瓶颈已经成为决定该应用成败的关键因素。这使得 Web 应用的性能测试受到许多研究人员和学者的广泛关注,也由此产生了许多相关的测试工具。但是这些工具大多使用传统的录制-回放技术,这种技术通过录制请求访问行为,形成反映客户端与服务器的端负载压力的性能测试脚本,接着通过模拟多用户并发执行测试脚本,得到 Web 应用系统的性能状态指标,如吞吐量、并发用户数、响

应时间等。但是这种简单录制测试操作人员请求并回放负载测试脚本的方式,仅仅反映了测试操作人员的操作特征,不能代表所有用户的行为,且脚本参数化掺杂着过多人工参与和不确定性。另一方面,测试脚本的简单回放只能保证给 Web 应用造成一定的并发负载,并不能说明实际用户的并发访问对 Web 应用产生的性能压力与此一致。面对互联网的高速发展,尤其是面对以分布性、异构性、平台独立性、用户群体巨大为特点的移动互联网的高速发展,这种过程掺杂过多的人工参与且仅代表测试操作人员单一操作的技术,难以确保测试结果的有效性。

为了解决这一个问题,应当使用一种能够充分利用真实用户的访问信息,通过从 Web 日志中提取用户会话数据,建立用户访问模式,生成并发测试脚本,用以辅助 Web 应用性能测试方法。在本文中我们提出了一种基于服务器日志中的

到稿日期:2013-09-26 返修日期:2013-11-16 本文受国家自然科学基金(61370103),广东省省部产学研重点项目(2012B091000050),广州市科技重大专项(2009A1-D041)联合资助。

梁力图(1990—),男,硕士,主要研究方向为软件自动化测试,E-mail:liang.lt@scut.edu.cn;陆 璐(1971—),博士后,博士生导师,主要研究方向为计算机远程监控、软件自动化测试以及图像异构计算,E-mail:lul@scut.edu.cn(通信作者)。

用户会话信息^[1]、结合经过序列匹配算法^[2]改进的凝聚层次聚类的算法来对 Web 应用的性能测试产生测试脚本的方法。相比于以往学者提出的用户会话聚类算法,我们所提出的算法的创新点在于考虑到用户会话中 URL 的序列性,在保持序列性情况下对用户会话进行聚类,使得产生的测试用例比不考虑 URL 的序列性的所产生的测试用例更有效性^[8]。该方法的基本模型如图 1 所示。

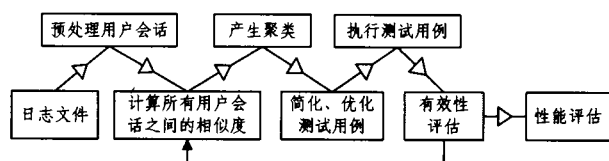


图 1 基于用户会话的 Web 应用性能测试模型

(1) Web 应用上线一段时间后,服务器中会记录这段时间内所有用户的请求日志信息,并从中抽取一部分日志文件进行预处理,包括去除用户不会显式请求的文件和用户请求访问失败的记录等。

(2) 从处理后的日志文件中提取用户会话信息。假设不同的用户具有不同的 IP,这样可以根据日志文件中的 IP 区分不同的用户。再利用 date 字段(根据用户会话一般为 30 分钟的法则),分离出完整的会话信息。

(3) 对提取出来的用户会话信息进行聚类。借用生物序列比对算法的思想,计算用户会话信息的相似度矩阵,再使用凝聚层次聚类算法根据相似程度对用户会话进行聚类。聚类之后得到的每一个类即是一类测试用例,每一类都是一种用户访问模式。并且可以得到每一类的大小(即每个簇中用户会话的数量)以及该类测试用例在整个测试套中的权重(即该类在所有用户会话中的权重)。

(4) 利用聚类后的会话信息生成测试脚本。首先从每类聚类中挑选相对比较完整的用户会话,使用其 URL 请求序列作为测试脚本中的请求序列;然后结合聚类结果,统计单位时间中的各个簇占总用户会话数的比例,得到各类测试用例的权重;最后结合需要模拟的用户规模,得到最终的测试集。

(5) 使用 LoadRunner 等回放工具回放测试集,产生大量的虚拟用户从而对 Web 应用产生负载,分析 Web 应用的性能状况。

本文第 2 节回顾前人所做的一些工作;第 3 节讨论针对 Web 应用的性能测试,如何根据 Web 应用的结构对 Web 应用的 URL 进行预处理,然后给出计算两个 URL 之间相似度的定义;第 4 节使用一种序列匹配算法并设计一个评分系统来计算用户会话之间的相似度,并将得到的相似度作为簇类之间的距离来进行层次聚类;第 5 节就如何在每个用户会话簇中选择典型的用户会话、去除冗余会话这一问题进行讨论,并给出一种方法来实现简化测试用例集。

2 国内外研究进展

到目前为止,Web 应用的性能测试的主流方法有以下两种:手写脚本方法和录制-回放方法。

在大多数的情况下,由于 Web 应用的复杂性和自动化测试工具的稀缺,性能测试脚本生成还是由测试人员手工撰写的。每一个虚拟用户执行一段子程序,子程序中不同的参数代表虚拟用户的不同行为^[3,4]。手写脚本是最灵活的,这种

方法几乎可以在 Web 应用的性能测试中都通用。但是,这种方法需要耗费大量的测试人员的人工工作,对于软件自动化测试的发展是不利的。

录制-回放技术也被广泛应用于软件自动化测试中。举例来说,由惠普公司编写的自动化性能测试软件 LoadRunner^[5]就是使用录制-回放技术。LoadRunner 首先记录下真实用户的操作,并将这些操作制作成测试脚本,随后通过大量的虚拟用户来对应用进行回放,进而对应用产生负载。类似的工具大多拥有友好的用户界面、强大的产生脚本的能力。使用这些工具通常不需要很强的编程能力,但是其缺点我们在上面也提到过,录制-回放技术仅能代表测试人员单一的操作且测试人员需要耗费过多的时间录制脚本以及设计参数。

事实上,已经有学者提出使用用户会话来辅助 Web 应用的性能测试。在文献[6]中,一种使用服务器日志中用户会话信息来辅助进行 Web 应用性能测试的方法被提出,文中讨论了基于用户会话的 Web 应用性能测试的发展前景以及和传统测试方法的区别,并提出一套基于用户会话的性能测试方法的流程模型,但是文中并没有具体地提到如何对用户会话信息进行聚类、简化和优化,只是提出了一套模型。Xiu Xia, Quan 等^[7]提出了基于会话的负载测试用户行为元模型,该元模型挖掘生产机环境下服务器日志自动产生的用户请求路径和用户请求数据来重构用户行为以协助 Web 应用进行性能测试。该工作应用决策树算法自动产生会话请求数据,以及利用最长公共子序列算法选择会话请求序列,但是他们并没有使用聚类算法对用户会话进行聚类,仅仅是使用最长公共子序列算法对用户会话进行简单的处理,算法时间复杂度和空间复杂度较高,且没有进一步阐述产生的用户行为如何协助重构测试负载对 Web 应用进行测试得到性能测试结果。

3 用户会话相似度计算

3.1 定义 URL 相似度

在聚类之前,第一个需要解决的问题就是如何定义两个会话之间的相似度问题,如果无法准确定义会话之间的相似度,则无法通过使用距离矩阵进行聚类分析。很多关于用户会话测试的文献中常用的一种方法是使用欧几里德距离来衡量两个会话之间的相似度,即把用户会话看作是一个 URL 的集合,通过计算两个会话之间 URL 相同的个数来衡量相似度。但是这种方法显然是有缺陷的:(1)使用欧几里德距离衡量相似度的时间复杂度和空间复杂度相当高;(2)用户会话中的 URL 请求实际上是用户在时间顺序上对 Web 应用的一系列请求操作,应该保证其序列性,应将用户会话中的 URL 看作是一组时间顺序上的一串序列,而并非是一组 URL 集合;(3)欧几里德距离不适合计算分类数据的相似性^[8]。

综合上述原因,本文将使用序列比对算法代替欧几里德距离来衡量用户会话相似度,以确保会话的序列性。实际上,序列比对算法最早使用在生物领域中,被用来比较两条生物序列的相似程度。目前存在不少关于生物序列比对的算法,包括 Needleman 算法^[9]和 FastLSA^[10]算法等,本文借用了其中的思想,结合 URL 序列进行了相应的改进。

用户会话之间的相似度计算分为两步:(1)定义计算 URL 之间相似度;(2)根据 URL 的相似度计算结果再计算用户会话相似度。

随着互联网技术的高速发展,现在的 Web 应用也越来越复杂,通常一个 Web 应用都会通过一个目录树将页面等资源放入,所以每个资源的 URL 都是有一个层次关系的,我们可以利用这种层次关系来对页面之间的相似度进行比较。

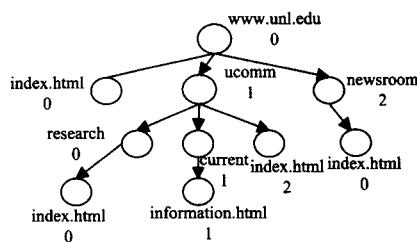


图2 截取某 Web 应用目录树一部分

为了比较某两个页面之间的相似度,首先我们给目录树的每一层的所有节点赋予一个标记,最左儿子节点赋予 0,其他依次类推。比如 URL 为 `www.unl.edu/ucomm/current/information.html` 所对应的标记为 0111,URL 为 `www.unl.edu/newsroom/realeases/publications.html` 所对应的标记为 0200。Web 应用中的每一个资源都能够在树中找到相应的标记值。

根据 URL 的标记值,我们可以通过以下两步来计算两个 URL 之间的相似度。

步骤 1 当比较两个 URL 时,从最左边开始逐个比较它们相应位置上的标记,直到出现第一个不相同的为止,记录下相同标志的个数,记为 N 。

步骤 2 假设两个 URL 中最长 URL 所拥有的标记数为 M ,给具有较长标记的那个 URL 的标记的每一位赋予一个权值,最后一位赋予权值 1,倒数第二位赋予权值 2 等。两个 URL 相同的位的权值的和除以所有位的权值的和则是两个 URL 的相似度,见式(1)。

$$\text{sim}(url_a, url_b) = \frac{\sum_{i=1}^M i}{\sum_{j=1}^M j} \quad (1)$$

3.2 定义用户会话相似度

在定义了如何比较两个页面之间的相似度之后,我们就可以通过比对 URL 序列并找到最优匹配的方法来比较两个用户会话之间的相似度,此部分的目的是通过序列比对算法找到一条使得两个 URL 序列最优匹配的比对路径。目前大多数序列比对算法通常建立在动态规划算法基础之上,这其中包括 Smith-Waterman(局部比对)算法和 Needleman-Wunsch(全局比对算法)算法。这些序列比对算法大多运用在论文中,Adrian Driga 等^[10]提出了一种比对生物序列,并找到最优匹配路径的方法。该算法可以把比对序列左移或右移并加上一个“gap”来寻求最优的匹配,确保最优路径的证明可在论文中找到,这里不再赘述,他们还提到可以通过一个评分系统来计算出序列的相似度。这些序列比对算法都运用于生物序列匹配中,然而,生物序列例如 DNA 序列、蛋白质序列都是非常长的序列,而字典规模相对较少。而对于 URL 序列来说,则恰好相反,URL 序列是相对较短的序列(一般为几十个到三百个 URL 链接)并具有相对较大的字典规模(几千个不同的网页资源)。这两者的匹配在时间、空间上的效率都是不一样的。我们在借鉴生物序列比对序列的基本思想后,提出一个简化的 URL 序列比对算法。

定义 1 每当通过左移和右移并加入 gap 或者两个页面的相似度为 0.0 时,我们给出一个罚分 (penaltyscore) - 10 分。当两个 URL 完全匹配时,给予一个奖励分 (postivescore) + 20 分。当相似度 x 的范围在 (0.0, 1.0) 之间时,给予的分数为 $x * (\text{postivescore} - \text{penaltyscore}) + (\text{penaltyscore})$; 当然两个 URL 之间的相似度与它们的序列长度有关。假设比对后所有得分的和为 totalscore ,则最终两 URL 的相似度定义见式(2)。

$$\text{sim}(US_1, US_2) = \frac{\text{totalscore}}{\text{Maxlength}(US_1, US_2) * \text{postivescore}} \quad (2)$$

定义 2 gap 可以插入到 URL 序列中的任何一个位置。

下面我们给出使用序列匹配来计算两个序列的相似度的具体方法。

假设有两个用户会话: US # 3 和 US # 4,它们包含的 URL 序列分别如下:

US # 1 = {1, 123, 126, 1, 2}, US # 2 = {1, 12, 123, 124, 12, 22}

	-	1	123	126	1	2
-	0(7)	-10	-20	-30	-40	-50
1	-10	20(6)	10	0	-10	-20
12	-20	10(5)	35	25	15	5
123	-30	0	30(4)	50	40	30
124	-40	-10	20	45(3)	55	45
12	-50	-20	10	35	55(2)	45
22	-60	-30	0	25	45	65(1)

图3 US # 1 和 US # 2 序列匹配过程

在比较两个 URL 序列时,先将其中一个序列放在矩阵的上方,另一个序列垂直放置在矩阵的左方。每一个序列的开始位置放置一个 gap 作为比较的开始点。寻找最优的匹配的过程实质上就是寻找从左上角作为起点到右下角作为终点的最佳路径。在寻找最佳路径的过程中,每一步只能向下移动,向右移动或者向右下对角线方向移动。这 3 种不同移动方式代表着 3 种不同匹配。

(1) 向对角线方向移动:说明对应的两个 URL 能够匹配,不需要添加 gap。

(2) 向下方向移动:说明当前的两个 URL 不匹配,在矩阵上方水平方向的 URL 序列需要在相应的位置添加一个 gap 以匹配垂直方向上的 URL 序列。

(3) 向右方向移动:说明当前两个 URL 不匹配,在矩阵左方垂直方向的 URL 序列需要在相应的位置添加一个 gap 以匹配水平方向的 URL 序列。

用以上的方法便可以分析出例子中 US3 和 US4 的最佳匹配方法为:

US3 1 - 123 126 1 2
US4 1 12 123 124 12 22

相似度得分为 65 分。当然两个序列的匹配与最长的那个序列的长度有关。所以根据上面提到的计算相似度的方法得到这两个会话的相似度为 0.65。

4 用户会话聚类

在计算会话之间的相似度之后,便可以通过凝聚层次聚类算法来对用户会话进行聚集。凝聚层次聚类算法是种自底向上的策略,首先将每个对象作为其簇,然后把这些原子簇合并为越来越大的簇,直到所有的对象都在一个簇中或者某个终止条件被满足。

层次聚类算法相比于另一种在数据挖掘中也经常使用到的划分方法,其优点在于并不需要用户提供聚类的数目 K 作为输入。它是一个从底向上的聚类算法,工作原理如下。

与一般的层次聚类算法不同的是,我们不使用欧几里德距离、cosine 度量或 Jaccard 函数来衡量簇类之间的距离,而是使用在前文提到的相似度作为度量用户会话簇之间距离的工具,这种度量方法将用户会话看作 URL 序列而不是 URL 集合,保证了用户会话的序列性和测试过程的有效性。

下面给出这个基于用户会话相似度的改进凝聚聚类算法步骤。

Step 1 采用上文提到的用户会话相似度算法,计算每一个会话之间的相似度,产生一个相似度矩阵,将每一个会话当成是一个簇,矩阵中的数据是两个簇之间的相似度,相似度越大,簇的距离越小。相似度与距离的关系为:

$$d(US_i, US_j) = |1 - \text{sim}(US_i, US_j)|$$

Step 2 将相似度最大(距离最小)的两个簇连接起来形成一个簇。采用平均距离公式来计算新产生的簇与其他簇之间的距离,平均距离公式为:

$$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{u \in C_i} \sum_{u' \in C_j} |US - US'|$$

Step 3 当产生的所有簇的距离都大于一个预先给定的阈值时,结束计算,否则回到 Step 2 继续进行聚类。

Step 4 输出聚类结果。

5 方法有效性验证

为了验证本文方法的有效性,我们必须比较并确认由本方法产生的虚拟用户行为和原本的真实用户行为是否具有一致性(即两方法在对 Web 应用造成的负载情况上是否一致^[11])。本实验使用了工业级测试工具 LoadRunner 来进行测试用例回放,并确认测试结果是否具有一致性。

在本仿真实验中采用的 Web 应用是基于 C#, .Net Pet Shop 4.0^[12],使用的是微软的 IIS6.0 服务器软件和 SQL Server 数据库。该 Web 应用的主要模块包括主页、注册页面、登录页面、购物车、许愿表等。数据库中包含了记录宠物信息、注册信息、交易数据和其他数据的 12 张表,并且使用了 JavaScript 和 Cookie 提供身份认证和个人服务。该应用包含了 5 个类型的 20 种宠物。实验室的网络环境如下:服务器端配置为:windows server2003, intel core i5, cpu 主频 2.27GHz * 2, 内存 4G。测试机配置为:win7, intel core i5, cpu 主频 2.27GHz * 2, 内存 2G。

第一个测试方法是由原始的用户会话中的请求作为测试脚本(即真实的网络负载情况)来对 .Net Pet Shop 进行测试;第二种测试方法则是采用本文提出的基于用户会话聚类的性能测试方法。通过确认两种方法分别在吞吐量和用户点击率是否具有相似性来证明本文方法的有效性。在使用 Load-

Runner 对两种测试方法生产的测试脚本进行回放后,得到的测试结果对比图如图 4 和图 5 所示,图 4 是两种方法对应的 Web 应用吞吐量对比图,图 5 是 httphits 对比图。

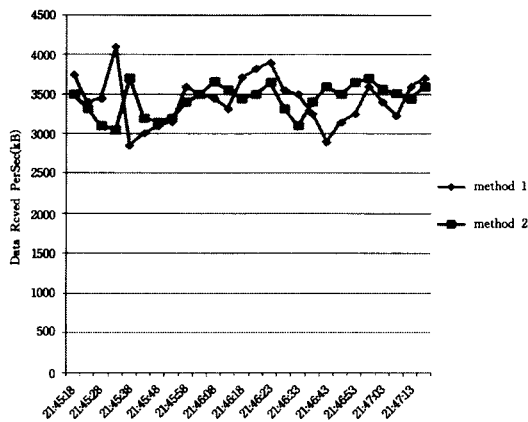


图4 两种方法对应的 Web 应用吞吐量对比图

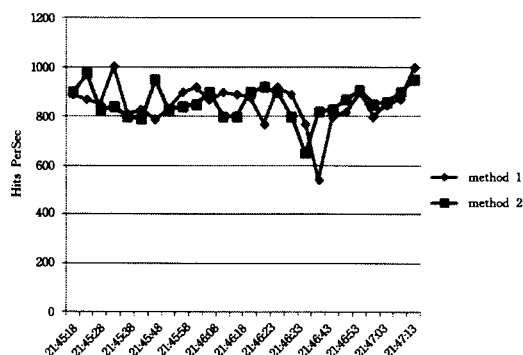


图5 两种方法对应的 Web 应用 httphits 对比图

从上述的对比结果可以看出,两次实验对 Web 应用造成的吞吐量稳定在 3500KB 左右,而用户点击率则稳定在 850 左右,且两次实验的曲线走势都比较吻合,这说明利用用户确实能够辅助性能测试。本文提出的基于用户会话的性能测试能够模拟出真实用户的行为,在性能测试上具有有效性。

结束语 在互联网技术飞速发展的时代,Web 应用的性能测试需求越发明显,一个互联网应用的性能决定着自身是否能够成功。为了保障 Web 应用的可靠性和质量,需要使用真实和有效的性能测试方法。但是传统的录制回放方法仅仅反映了测试人员单一的操作特征,且脚本参数掺杂了过多的人工参与。面对功能日益复杂、用户数量日益增多的 Web 应用,传统的性能测试方法产生的大量模拟用户可能与实际用户的行为有很大的差别,导致性能测试结果的有效性受到质疑。如何利用真实的用户信息辅助性能测试成为了近年来测试方向的新趋势。

针对已经上线并开始运营的 Web 应用(具有一定数量的服务器日志文件),本文提出一种新颖的基于用户会话的性能测试方法。该方法结合改进的凝聚层次聚类算法,借助服务器日志中记录的用户会话信息辅助自动化产生测试用例以进行性能测试。其主要过程为:先对 Web 应用的服务器日志进行清洗和处理,分离出用户会话信息;计算各用户会话之间的相似度,构建用户会话距离矩阵;利用上一阶段得到的用户会话距离矩阵进行聚类分析,得出不同的用户访问模式;通过用户访问模式和聚类信息产生测试用例集。第 3 节的对比实验

(下转第 68 页)

逼真地模拟现实用户关系。

参考文献

- [1] <http://developer.android.com/guide/components/intents-filters.html>
- [2] 孟祥武,胡勋,王立才,等. 移动推荐系统及其应用[J]. 软件学报,2013,24(1):91-108
- [3] Woerndl W, Schueller C, Wojtech R. A hybrid recommender system for context-aware recommendations of mobile applications [C]// 2007 IEEE 23rd International Conference on Data Engineering Workshop. IEEE, 2007: 871-878
- [4] Yan Z, Zhang P, Deng R H. TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applications[J]. Personal and Ubiquitous Computing, 2012, 16(5): 485-506
- [5] Shi K, Ali K. GetJar mobile application recommendations with very sparse datasets[C]// Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2012: 204-212
- [6] Yan B, Chen G. AppJoy: personalized mobile application discovery[C]// Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. ACM, 2011: 113-126
- [7] Davidsson C, Moritz S. Utilizing implicit feedback and context to recommend mobile applications from first use[C]// Proceedings

- of the 2011 Workshop on Context-awareness in Retrieval and Recommendation. ACM, 2011: 19-22
- [8] Bray T, Paoli J, Sperberg-McQueen C M, et al. Extensible markup language (XML)[J]. World Web Journal, 1997, 2(4): 27-66
- [9] Fallside D C. XML Schema Part 0: Primer [EB/OL]. <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>, 2001
- [10] Thompson H S. XML Schema Part 1: Structures [EB/OL]. <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>, 2001
- [11] Biron P V. XML Schema Part 2: Datatypes [EB/OL]. <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>, 2001
- [12] Jannach D, Zanker M, Felfernig A, et al. Recommender systems: an introduction. Cambridge University Press, 2010
- [13] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 1998: 43-52
- [14] 吴军. 数学之美[M]. 北京: 人民邮电出版社, 2012
- [15] Brown P F, Cocke J, Pietra S A D, et al. A statistical approach to machine translation[J]. Computational linguistics, 1990, 16(2): 79-85
- [16] Rabiner L, Juang B. An introduction to hidden Markov models [J]. ASSP Magazine, IEEE, 1986, 3(1): 4-16
- [17] 项亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012

(上接第 49 页)

表明,本文提出的测试方法确实能够模拟出真实用户对 Web 应用造成的负载,证实了本方法的有效性。

相比于传统的录制回放测试方法以及其他的基于用户会话的测试方法,本文提出的方法具有以下 3 个优点:(1)该方法能够自动生成性能测试用例,省去录制和设计测试脚本的工作;(2)借助真实用户的用户会话信息,可以使测试脚本产生的模拟负载与真实负载相一致,保证了性能测试过程的真实性和测试结果的有效性;(3)相比于其他的基于用户会话的测试方法,本文方法使用了生物序列比对算法,将用户会话中的 URL 请求看作是一组序列而不是一组集合,保障了用户会话的序列性。

未来的研究工作将会集中在以下几个方面:(1)增强对无效日志文件的识别;(2)对用户会话相似度算法进行优化改进,降低算法和空间复杂度;(3)挖掘更多的日志信息,例如针对移动互联网应用,日志信息提供的浏览器信息可以把不同品牌的手机用户进行区分(不同的浏览器对移动互联应用发出的请求,服务器端返回的数据将有所不同),产生不同的测试用例;(4)研究基于用户会话的性能测试方法的性能评估。

参考文献

- [1] Elbaum S, Rothermel G, Karre S, et al. Leveraging User-Session Data to Support Web Application Testing[J]. IEEE Transactions on Software Engineering, 2005, 31(3): 187-202
- [2] Myers E, Miller W. Optimal alignments in linear space [M]. Computer Applications in the Biosciences (CABIOS). 1988: 11-17
- [3] Rudolf A, Pirker R. E-Business Testing: User Perceptions and Performance Issues[C]// Proceedings of the First Asia-Pacific Conference on Quality Software. IEEE Press, 2000: 231-239

- [4] Subraya B M, Subrahmanya S V. Object Driven Performance Testing of Web Applications[C]// Proceedings of the First Asia-Pacific Conference on Quality Software. IEEE Press, 2000: 117-125
- [5] Mercury Interactive Corporation. Load Testing to Predict Web Performance[R]. Technical Report WP-1079-0604. Mercury Interactive Corporation, 2004
- [6] Liu Yue, Wang Kang, Wang Wei, et al. User-session-based Test Cases Optimization Method based on Agglutinate Hierarchy Clustering[C]// IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing. 2011: 414-418
- [7] Quan Xiu-xia, Lu Lu. Session-Based User Behavior Meta-Model of Web Applications for User-Level QoS Load Testing[C]// Supply Chain Management and Information Systems (SCMIS). 2010: 152-167
- [8] Wang Wei-nan, Zaiane O R. Clustering Web Sessions by Sequence Alignment[C]// Proceedings, 13th International Workshop on Database and Expert Systems Applications, 2002. 2002: 394-398
- [9] Needleman S, Wunsch S. A general method applicable to the search of similarities in the amino acid sequence of two proteins [J]. Journal of Molecular Biology, 1970: 443-453
- [10] Adrian D, Pau L, Jonathan S, et al. FastLSA: A Fast, Linear-Space, Parallel and Sequential Algorithm for Sequence Alignment[C]// Proceedings of the 2003 International Conference on Parallel Processing. 2003: 304-309
- [11] Diwakar K, Jerome R, Shikharesh M. A Synthetic Workload Generation Technique for Stress Testing Session-Based Systems [J]. IEEE Transactions on Software Engineering, 2006, 32(11): 868-882