基于统计数据的 Web 站点测试*)

许 蕾 徐宝文 陈振强

(东南大学计算机科学与工程系 南京 210096) (武汉大学软件工程国家重点实验室 武汉 430072)

(江苏省计算机信息处理技术重点实验室开放基金 苏州大学 苏州 215006)

Testing Website Based on Statistic Data

XU Lei XU Bao-Wen CHEN Zhen-Qiang

(Department of Computer Science & Engineering, Southeast University, Nanjing 210096)

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)

(Jiangsu Province Key Laboratory of Computer Information Processing, Soochow University, Suzhou, 215006)

Abstract It is not practical to test a website exhaustively because of too many pages and user groups. So we begin our work at the log files of the Website server, gathering the users' visits and the server's responses in a period of time. Then we analyze and discuss these information, and based on this, we identify the key pages, predominate pages and users' visiting modes. Thus we can allocate the testing resource appropriately, emphasing on these pages and modes. Finally we may improve the display structure of the site and fulfill the functionality of the site, enhancing users' visiting efficiency as possible as we can.

Keywords Website testing, Statistic data, Key page, Predominate page, Visiting mode

1 引言

Web 应用具有广泛性、交互性和易用性等特点，其迅猛的发展态势吸引了人们广泛的关注[19]。能够吸引尽可能多的用户并对其长时间关注是网站追求的主要目标，也是衡量一个网站是否成功的主要指标[17]。要实现这样的目标，就需要从用户的角度出发设计、实施、调整整个网站的结构和内容，同时要通过测试来保证网页功能的正确性、有效性和完善性。由于 Web 具有分布、异构、并发和平台无关的特性，因而 Web 测试要比普通程序的测试复杂得多[6,11]。

传统的测试方法包括白箱测试、黑箱测试等方法[3]，其测试原则都是要达到尽可能高的覆盖率，也即通过执行适当的测试用例来了解原程序的运行情况。由于程序拥有数目众多的变量、函数以及分支、循环、顺序等多种结构，再加上各种情况的组合，使得测试时不可能覆盖到所有的组合状况，甚至要达到一定的测试覆盖率也很困难[1,13,14]。对于 Web 站点来说，页面数量众多，可以通过链接指向到任意 URL，与用户的交互也比较多，在这样复杂的环境下要进行全面的测试是不太可能的，因而合理分配测试资源显得比较重要。

目前，国内外已开始对 Web 测试进行研究并取得了一些初步研究成果[2,4,5,6,9]，本文的目的是探讨如何进行有针对性的测试。为此，本文首先用网状图来表示站点的基本结构，接着通过分析 Web 站点服务器端的日志文件得到一些统计数据，从而得到加权网状图；在此基础上，定义了关键页面、必经页面和用户访问模式；然后详细讨论了相应的测试工作和测试技术；最后是对未来工作的展望。

2 基于统计数据的 Web 站点结构表示

2.1 Web 站点基本结构

Web 站点是由众多的 Web 页面组成的，Web 页面按其内容的生成方式通常分为两种：如果其内容是固定不变的，就称其为静态页面；如果其内容是在运行过程中动态生成的，则为动态页面。用户通过浏览页面来获取自己所需的信息。页面不是孤立存在的，而联系它们的纽带通常就是超级链接，沿着超链可以很方便地从一个页面跳转到另一个页面。众多的页面通过链接联系在一起就组成了一个错综复杂的网状结构。为了方便描述，我们给出一个简化的 Web 站点结构图[16]。

定义 1 在简化 Web 站点结构图中，称 Web 页面为页面节点，用二元组 Page=(P_id,P)表示，P_id 唯一标记一个页面节点；P 为属性集，P={p_i|p_i 为属性，i=1,2…}，属性可以是相对 URL、类型、链接点集合、容器、内容、修改日期等。

根据 Web 页面中链接存在与否及链接的指向，页面节点可以分为孤立页面节点、源页面节点和目标页面节点。孤立页面节点是不包含任何链接的页面节点；包含链接的页面节点为其中的链接的源页面节点；链接所指向的页面节点叫做该链接的目标页面节点。很显然，页面节点相对于不同的链接而言，既可以为源页面节点，也可以是目标页面节点。

定义 2 页面中的链接用四元组 Link=(L_id,string,source_id,target_id)表示，其中，L_id 唯一标记一个链接点，string 描述了该链接的显示信息，source_id 为链接所在的页

*)本研究得到国家自然科学基金(60073012)、江苏省自然科学基金(BK2001004)、江苏省科技攻关项目(BE2001025)、教育部高校骨干教师资助计划、江苏省三三三人才基金、高等学校重点实验室访问学者基金、武汉大学软件工程国家重点实验室开放基金资助、江苏省计算机信息处理技术重点实验室开放基金(苏州大学)资助。许 蕾 博士研究生，主要从事 Web 软件分析、测试与应用方面的研究工作。徐宝文 教授，博士生导师，主要从事程序设计语言、软件工程、知识与信息获取技术等方面的教学与科研工作。陈振强 博士研究生，主要从事软件分析、理解与测试方面的研究工作。

面节点，target_id 为目标页面节点。

定义 3 整个 Web 站点结构图用二元组 G=(N,E)表示，其中 N={Page}，E={Link}。

对一个链接而言，其指向有很大的随意性，即该链接的目标页面节点可以为任何页面节点，因而在一张 Web 站点结构图上就有环路存在的可能。所谓环路就是说，从某个页面节点出发的链接经过多个(包括 0 个)页面节点传递后，最终有链接指回到该页面节点，这些链接就组成了一个环路，其中不经过任何页面节点传递的链接构成的环路称之为自环。

另外，如果一个链接的源页面节点不属于该 Web 站点，即这是外部世界指向该站点的一个链接，称之为外部指向链接；相应地，如果一个链接的目标页面节点不属于该 Web 站点，即通过该链接可以跳转到外部世界，称之为指向外部链接。这两种链接统称为外部链接，它们在一定程度上构成了该 Web 站点的入口和出口。

为了简化模型、突出重点，我们主要研究一个站点内部的页面之间的直接联系情况，对环路(包括自环)和外部链接的情况不做考虑，这样我们的研究对象就是 Web 站点内部的各个页面以及页面上的链接，或者更明确地说就是选取一个页面，重点研究该页面的内容以及其入度和出度情况(入度指所有目标页面节点为该页面的链接的集合，出度是指该页面上所有链接的集合)。

2.2 获取统计数据

Web 应用的最终用户的看法是评判 Web 应用好坏的标准。为了找到用户认为重要的页面，需要对用户的行为进行统计分析。在服务器端有日志文件，记录用户对站点的具体访问信息，因而可以对日志信息进行分析，通过数据挖掘得到用户访问页面的具体情况，从而根据这些历史信息来预测用户的未来行为，找出存在的以及可能的关键页面。日志文件包括：请求页面机器的 IP 地址，请求发出的日期和时间，用户请求的页面，与请求相关的 URL，用户上一次访问的页面，所请求文件的大小，HTTP 状态码[2]，其日志记录格式表示为：

〈req_IP，req_D&T，req_page，ref_URL，lastv_page，file_size，HTTP_status〉。

一般而言，在日志记录中 ref_URL 和 lastv_page 不会同时出现，这就需要做一些额外的查找匹配工作。例如，如果仅出现 lastv_page，要推导出 ref_URL，就需要遍历 lastv_page 中的超链接，即如果存在某个链接 e，e.source_id 为 lastv_page 且 e.target_id 为 req_page，那么 e 就是 ref_URL。为了简化起见，我们所分析的日志记录事先都经过了上述处理，符合以上定义的格式。

通过对一段时间内的日志文件进行统计分析，可以发现用户对该站点内页面的访问次数以及链接的点击率，从而可以发现关键页面和用户的访问模式，在此基础上优化调整 Web 站点结构，合理分配测试资源，做到统筹兼顾、重点突出，最终使之满足用户的需求、提高用户的访问效率。

统计所得的主要相关数据为站点内各页面的访问次数(visit_counts)和各链接的点击次数(hit_counts)，这只需要选出 HTTP_status 值为正常的记录，然后对 req_page 项中值相同的记录分别累加，这样就得到各个页面的访问次数；类似地，对 ref_URL 项中值相同的记录分别累加将得到各个链接的点击次数。进一步地，如果结合 req_D&T 考虑，就可以得到不同时间段的服务器响应情况，从而可以进行站点压力和强度等性能分析；如果再考虑 req_IP 的因素，可以对用户的分布情况有更深入的了解，从而为提供个性化服务或者站点镜像打下基础；当然，文件的大小 file_size 也是影响页面访问成功与否的重要因素，通常较大的文件需要更多的传输时间，一旦用户等待时间太长，用户往往就会选择放弃访问。

2.3 生成加权结构图

这样，在前面建立的站点结构图上，我们就可以为节点和边分别添上各自的权值：visit_counts 和 hit_counts。通常 visit_counts 值较大的页面具有更高的重要性，一方面可能是其内容比较重要，吸引了更多的用户；另一方面也可能是其在结构图中的位置比较重要，是其它页面的必经页面。这样，在确定了阈值后就可以识别出关键页面和必经页面。

为了比较各链接的重要程度，我们将综合一个页面节点上的所有超链的点击情况，得到各链接的相对点击率 hit_rate，即对某个链接 e 而言，其

$$hit\_rate = e.hit\_counts / (\Sigma\, l.hit\_counts)$$

其中，l.source_id=e.source_id。

hit_rate 值较大的链接通常比较重要，另外也可以在这张加权结构图上发现用户的访问模式，即用户访问页面的先后顺序，从而可以找出一些潜在的规律，进一步调整站点的结构，如将一些相关程度较高的页面组织在一起，或者通过修改链接来缩短用户的访问路径。这样，在基于统计信息的 Web 站点结构图的基础上，我们就可以进行有针对性的、有效的测试。

关键页面和必经页面的重要性决定了要对其进行重点测试，在测试资源有限的情况下要优先对其进行测试，测试的内容包括功能、性能、可用性和兼容性等。这些重要页面通常为站点首页或者是与用户有交互的页面，要求进行更详尽、更细致的测试。

3 针对关键页面的可用性测试

3.1 页面可用性要求

站点首页是站点的门户，为了给用户留下深刻的第一印象，必须有其独特的方面：有的以内容丰富取胜，有的则以功能强大著称，同时也要在版面布局、色彩安排上精心考虑。这里重点讨论可用性方面的要求。可用性评判具有较强的主观性，因为不同用户的喜好往往有很大的不同，很难找到一个统一的标准。不过在综合大多数用户的反映情况以后，我们还是可以得到一些普遍意义上的要求的。结合自己的实际使用经验和用户的普遍要求，我们发现：简洁、一致、对比度好的页面很受用户欢迎[10,12]。

站点首页要为人们所理解和接受，必须以满足人们的实用和需求为目标。从人记忆能力的角度来说，由于人的大脑一次最多能记忆 5 到 7 条信息，因而最好先用几个简单的关键词语或图像来吸引用户的注意力，然后再合理安排各项内容，而不是大量庞杂内容的堆砌。一个网站在风格上应该尽量保持一致，一方面能够体现该网站的特点，使之与其它网站区分开来，另一方面能够提供格式相同的导航，大大简化了用户的使用。合理的对比能够突出重点，强化概念，使内容更易于辨认和接受，从而给用户留下深刻的印象。

3.2 页面可用性测试方法

从上述要求出发，我们可以制定出站点页面可用性测试的具体方案，以得到简洁、一致、对比度好的页面，具体的测试方法如算法 1 所示，从页面素材、内容的检查测试到页面布局的合理性测试，再到页面对比效果测试和风格一致性测试，测

试条理清晰，测试内容全面。

测试时我们注意到，如果页面访问量大，而页面上的链接点击率低，则意味着用户对该关键页不太满意，放弃了进一步的访问。

算法 1 Web 站点面页可用性测试方法及步骤

1 页面素材内容的测试

1.1 检查页面文字输入的正确性，包括错别字、拼写、语法等内容的检查，一旦发现错误立刻进行修正。

1.2 检查页面图片的简洁性，确保图片有明确的含义和用途，同时限制所用字体和颜色的数目（每页使用的字体不超过 3 种，使用的颜色少于 256 种）。

2 页面布局合理性的测试

2.1 检查页面内容的组织方式，看其是否为按逻辑、按时间顺序、按地理位置或其它合理方式中的一种来组织页面内容，如果组织方式比较凌乱，需要进行适当调整。

2.2 检查内容的相关性，把相关的内容放在一起，而不相关的内容用空白、水平线或其他图形分隔开。

2.3 检查内容的重要程度，把希望浏览者最先看到的内容放在页面的左上角和页面顶部，然后按重要性递减的顺序，由上而下放置其它内容。

2.4 检查链接放置位置的恰当性，在段落中不宜放入过多的链接，最好按逻辑关系放置，可以使用框架，一边放置链接，另一边显示文本，提示浏览者去使用这些链接。

3 对比效果测试

3.1 检查颜色上的对比效果，一般内容提要和正文使用不同颜色的字体，文本和它的背景之间不宜采用太强的颜色对比。

3.2 检查字体的变化，可以使用斜体和黑体突出关键内容，但不能滥用。

3.3 检查图像与文本的对比，注意不宜使用太多的图像。

4 页面风格一致性测试

4.1 检查版式的一致性，要求各个页面使用相同的页边距，文本、图形之间保持相同的间距，主要图形、标题或符号旁边留下相同的空白。

4.2 如果使用图标导航，则各个页面应当使用相同的图标。

4.3 页面中每个元素与整个页面以及站点的色彩和风格要保持一致。

4.4 文字的颜色要同图像的颜色保持一致并注意色彩搭配的和谐。

4 针对必经页面的交互功能测试

在一般的 Web 站点中，用户可以访问 Web 站点上的任何信息，且访问信息的次序是不受限制的。但是对于某些特定的应用，如基于 Web 的电子邮件系统、基于 Web 的需要进行用户口令验证的 Intranet 系统等，必须限制用户的访问信息的次序：它要求用户首先在登录页上输入登录信息，然后根据用户名动态生成相应的主页。这样，测试的内容是两个有着先后关系的页面，不同于对单个页面进行的测试[15]。前一个页面是后一个的必经页面，需要重点进行功能性和安全性等方面的测试。

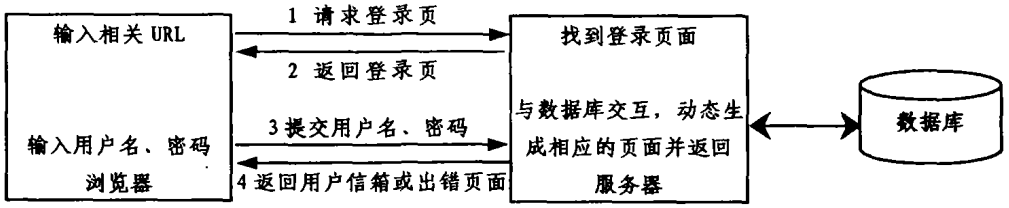

图 1 页面交互流程示意图

图 1 显示了用户使用个人信箱的基本流程，过程如下：用户首先输入站点的 URL 地址，请求服务器返回站点登陆页面；然后提交用户名和密码到服务器端；服务器端的脚本程序在获取用户名和密码以后，将其与存储在数据库中的相关信息进行比较对照；如果均符合条件，将返回该用户的信箱页面，否则返回的是提示输入有误的页面。

由于这些操作主要是由这个脚本程序完成的，因而必须对其功能进行全面的测试，包括信息获取功能、查询匹配功能和信息返回功能等。另外，对页面显示的先后顺序还要进行测试，不输入用户名和密码就不能进入到下一个页面，这也是“必经页面”的意义所在。

功能测试通常采用黑箱方法，与传统软件的测试方法没有太大差别，一般是根据规格说明生成测试用例，执行用例，将用例的输出与期望输出比较，从而发现软件功能上存在的问题。这里比较特殊的是对页面显示顺序的测试，文[15]根据页面间变量之间的联系生成页面状态图，给出了对页面显示先后顺序进行测试的方法。

有时为了减轻服务器端的压力，可以在浏览器端设置一个小 JavaScript 程序，通过它来保证用户不输入非法字符。比如说，当用户输入的信息不符合要求时，将直接弹出报错窗口，而无需将信息提交到服务器端然后再返回出错信息。这样就过滤掉了一些信息噪声，减轻了服务器和网络传输的压力。这个 JavaScript 程序中包含了一些对输入字符串的约束条件，通过对照就可以判断了。当然，事先应该确保该程序的功能。

黑客攻击网站往往需要首先获取用户名和密码，因而必经页面的安全性显得至关重要。安全性测试的具体方法与功能测试基本类似，不再赘述。这里重点讨论信息传送过程中的安全性问题。信息从客户端提交到服务器端的方式通常有两种：GET 和 POST[7]。GET 方法指示浏览器把查询值作为 URL 的一部分，多个域值归并成一个查询字符串，每个域值用“&”符号隔开，这样的信息很容易被截取或修改。例如，以前“5460 网站”的留言系统就存在这样的问题，任何人都可以在 HTML 源文件上修改或删除留言人的姓名，将该文件存盘后，打开保存后的文件就可以发现留言人的姓名有了变化。POST 方法指示浏览器用命令文件的标准输出发送表格数据，而不是采用查询字符的方式，而且当 GET 方法受制于服务器环境变量空间大小或者当浏览器 URL 长度有限时，

POST 方法可以处理大量的数据，十分有用。命令文件的安全性有了一定的保障，但也不能说就没有漏洞，通常在发送重要数据时要作加密处理。

5 用户访问模式测试

前面的测试内容主要是针对单个页面以及两个有链接相连的页面的，相当于在站点结构图上进行零步或一步传递运算。为了进一步找出站点内多个页面之间的关系，还需要进行多步传递运算，从用户的角度来说，就是要找出用户访问模式。

定义 4 用户访问模式是一些页面和链接的集合，在这些页面中，用户只要访问了其中的一页，则可以断定他也要访问其它的网页，并且这些页面不一定有链接直接相连。

发现用户的访问模式具有较大的难度，因为用户的行为具有很大的随意性，就一两次访问行为而言，很难找到其中的规律。因而需要收集大量的用户访问信息，在综合了大量用户访问行为的基础上，去掉一些偶发事件，得到一些具有用户访问共性的模式。可以通过数据挖掘技术来发现用户访问模式，目前也已有了一些描述算法[18]。这些模式也就是一些从大量信息中提取出来的知识，对测试工作的开展具有积极的指导意义。例如，如果很多人具有 a. html→b. html→c. html 这样的访问模式，则我们可以认定 a. html 和 c. html 之间有一定的关系，可以考虑在 a. html 上直接加上到达 c. html 的链接。

这样，根据不同的用户访问模式，我们就可以把网页分组，得到一个一个的兴趣点，从而根据用户行为的相似性，把用户按行为模式分类。另外还可以根据用户访问模式来修改网页之间的连接，把用户想要的页面以更快且有效的方式提供给用户。总之，根据用户的访问模式，我们可以挖掘出页面之间的关联关系，调整站点结构，使之布局更合理、用户访问效率更高。

根据用户访问模式规则，通过聚类算法可以将加权 Web 站点结构图划分成一些模块，模块内部的页面内容相关、联系密切，而不同的模块就反映了不同的兴趣点。模块级测试的难度将有所下降，因为模块之间的联系不太密切，没有开展交互功能测试的必要，最多只需要进行链接可达性测试。这样，我们的测试工作就可以分组同步进行了：一组进行首页的可用性测试；一组进行模块内的交互功能测试；一组进行模块间的链接可达性测试，这样将在很大程度上提高测试的工作效率。同时，我们也可以对站点结构进行适当的调整，合并内容相关的页面，或者设置新的链接将两个原本要经过多次链跳转才相连的页面直接链在一起，这样页面的主题更突出、用户的访问更方便。所以说，在用户访问模式指导下的测试效率、测试效果都将有显著的改善。

结束语 随着 Web 应用种类的不断增加，作为保证 Web 质量和可靠性的重要手段，Web 测试受到人们越来越多的重视。由于 Web 页面和用户数量巨大，基于测试覆盖率的传统方法难以完成如此繁重的测试任务。为此，本文考虑从服务器端的日志文件中挖掘出一些有用的统计数据，在此基础上进行有针对、有侧重的测试。本文首先提出了一个简化的 Web 站点模型，这是一个加权的 Web 站点结构图；然后根据图上的信息得到关键页面、必经页面和用户访问模式，从而进行关键页面的可用性测试、必经页面的交互功能测试和划分模块后进行的分组测试。这样本文从一个独特的角度探讨了如何进行有效的 Web 站点测试，提出了一些切实可行的测试方法。在未来的工作中，我们将进一步通过实验来验证我们的理论结果。

参 考 文 献

1 Xu Baowen, Chen Zhenqiang. Dependence Analysis for Recursive Java Programs. ACM SIGPLAN Notices. 2001, 36(12): 70～76

2 Kallepalli C, Tian J. Measuring and Modeling Usage and Reliability for Statistical Web Testing. IEEE Trans. Software Engineering, 2001, 27(11): 1023～1036

3 Kung D, Gao J, Hsia P, Lin J, Toyoshima Y. Design Recovery for Software Testing of Object-Oriented programs. In: Proc. of Working Conf. on Reverse Engineering, May 1993. 202～211

4 Kung D C, Liu C-H, Hsia P. An Object-Oriented Web Test Model for Testing Web Applications. In: Proc. of APAQS 2000, Oct 2000. 111～121

5 Ricca F, Tonella P. Web site analysis: Structure and evolution. In: Proc. ICSM2000, 2000. 76～86

6 Gao J, Chen C, Toyoshima Y, Leung D. Engineering on the Internet for Global Software Production. IEEE Computer, May 1999. 38～47

7 Jamsa K et al. WEB 程序设计教程，电子工业出版社，1997

8 Warren P, Boldyreff C, Munro M. The evolution of websites. In: Proc. of the Intl. Workshop on Program Comprehension, May 1999. 178～185

9 Warren P, Boldyreff C, Munro M. Characterizing evolution in Web sites: Some case studies. In: Proc. of the Intl. Workshop on Web Site Evolution, Oct 1999

10 Software QA and Testing Frequently-Asked-Questions. http://www.softwareqatest.com

11 Powell T A, et al. Web Site Engineering: Beyond Web Page Design, Prentice Hall, 1998

12 Web SiteTesting. http://www.telsoft-inc.com

13 Chen Zhenqiang, Xu Baowen, Yang H. Detecting Dead Statements for Concurrent Programs. In: Proc. of SCAM 2001, IEEE CS Press. Florence, Italy, 65～72

14 Chen Zhenqiang, Xu Baowen. Slicing Concurrent Java Programs. ACM SIGPLAN Notices, 2001, 36(4): 41～47

15 卢虹，徐宝文．一种 Web 应用的状态测试方法．计算机工程与应用，2002，2：55～57

16 徐宝文，张卫丰．数据挖掘技术在 Web 预取中的应用研究．计算机学报，2001，4：430～436

17 许蕾，徐宝文，陈振强．Web 测试综述．计算机科学，已录用待发表

18 阳小华，周龙骧．基于用户访问模式的 www 浏览路径优化．软件学报，2001，12(6)：846～850

19 张卫丰，徐宝文，周晓宇，李东，许蕾．元搜索引擎研究．计算机科学，2001，28(8)：36～41