

Web 性能测试回放浏览器的设计、实现与比较^{*}

叶新铭 刘 亮

(内蒙古大学计算机学院 呼和浩特 010021)

摘 要 本文分析了回放浏览器在 Web 性能测试中的作用,提出了支持在线和离线两种回放模式的 Web 性能测试回放浏览器 BView 以及离线回放实现中的关键技术——Web 资源本地存储的设计与实现,还对 BView 的实现进行了验证。最后给出了一个 Web 回放模型,在 BView 同 Run-Time Viewer 进行功能对比的基础上,采用 Web 回放效果对两种 Web 回放浏览器进行了定量比较。

关键词 回放浏览器, 离线回放, Web 回放模型, Web 回放效果

Design, Implementation and Comparison of Web Performance Testing Replay Browser

YE Xin-Ming LIU Liang

(College of Computer, Inner Mongolia University, Huhhot 010021)

Abstract This paper analyzes the role of Replay Browser in Web performance testing, and puts forward the design and implementation of Web performance testing replay browser which supports on-line and off-line replay modes and Web resource local storage which is the key technology in implementation of off-line replay. Furthermore, it performs the validation of BView implementation. Finally, it puts forward a Web replay model, and makes a quantity comparison between these two Web replay browsers using WRE based on the function comparison between BView and Run-Time Viewer.

Keywords Replay browser, Off-line replay, Web replay model, Web replay effect

1 引言

随着 Internet/Intranet 技术及应用的不断成熟, Web 应用系统迅速发展。由于 Web 系统访问用户众多, 在上线前必须进行压力测试, 以模拟大量用户访问系统的真实情况, 发现、定位并解决系统瓶颈问题。手工压力测试有着严重的缺陷, 采用工具测试已经成为主流。但无论手工测试还是工具测试, 测试例对真实业务逻辑的正确反映则是保证压力测试的前提条件。一个错误业务逻辑的测试例的执行并不能准确模拟真实用户压力情况, 其测试结果可信性不高。

当前主流压力测试工具均采用了录制回放技术, 即用户根据业务逻辑将 Web 操作录制为脚本, 通过组织测试场景、创建虚拟用户执行录制的脚本实现压力测试, 而脚本对业务逻辑的准确反映则成为压力测试成功的关键环节。

在使用基于脚本录制回放的测试工具进行测试时, 几乎所有的测试工具使用者都期望, 在脚本回放执行器执行脚本时, 能够使用浏览器以图像化形式来显示 HTTP/S 协议 Web 响应, 因为图像显示模式相比文本显示模式而言, 更为直观和准确, 脚本的真实执行情况可以一目了然地展现在测试人员的面前。这种功能有助于对脚本进行建模, 因而许多压力测试工具都设计了回放浏览器进行脚本回放, 以有效地帮助测试人员调试录制的脚本。

2 回放浏览器设计与实现

2.1 回放浏览器简介

回放浏览器包括回放执行器和回放显示界面两部分, 回

放执行器用于执行虚拟用户脚本中的 URL 请求, 并将回放结果显示在回放显示界面上。回放浏览器不同于 IE 等一般 Web 浏览器: 一般 Web 浏览器向 Web 服务器发送的 URL 请求需要由用户通过键盘和鼠标输入, 而回放浏览器发送的 URL 请求则是直接从录制的虚拟用户脚本中读取。回放执行器根据读取的 URL 请求, 向操作系统发送 API 命令, 使操作系统重新完成与录制相同的 Web 请求动作, 从而实现了回放的效果。回放浏览器回放显示形式分为文本和图像两种, 文本形式只显示回放执行器向 Web 服务器发送的 URL 请求的文字信息, 而图像形式则将 URL 请求对应的 Web 页内容进行显示。显然图像形式更利于测试人员观察、比较回放运行情况。

在脚本回放时, 回放浏览器首先对虚拟用户脚本进行语法检查, 判断语法是否正确。如果存在错误, 则停止回放并提示语法错误。语法正确的情况下, 回放浏览器启动回放执行器, 逐条读取虚拟用户脚本语句中的 URL 请求, 通过网络并发送给 Web 服务器, 在接收到 Web 服务器的响应后, 将响应结果以图像或文字的形式显示在回放显示界面上。如果回放脚本执行结束, 则停止回放执行器完成回放过程。

2.2 在线和离线回放模式

一般情况下, 回放时需要连通网络, 因为回放执行器需要从 Web 服务器获取 URL 请求对应的 Web 页内容, 我们称这种回放模式为在线回放。在线回放模式下, 脚本回放显示的速度受当时网络环境和 Web 服务器响应速度的影响, 整个回放过程耗用的时间与录制时相当。尤其是在脚本中语句数较多的情况下, 进行一次回放耗费很长, 不便于测试人员频繁对

^{*} 国家自然科学基金项目(60263002)、内蒙古科技攻关项目(2002061002)。叶新铭 教授, 博士生导师, 主要研究方向: 计算机网络; 刘 亮 硕士研究生, 主要研究方向: 计算机网络。

修改后的脚本进行调试。

虽然在线回放获取并显示的 Web 页内容是最新的,但是当 Web 页内容相对稳定,而测试人员对回放速度有更高的要求或网络不能连通时,在线回放已经不能满足测试人员的要求。另一方面,由于一般的测试工具在脚本录制时并不在本地保存 URL 对应的 Web 页内容,导致测试人员在需要查看 URL 请求对应的 Web 资源内容时,只能通过重新向 Web 服务器发送 URL 请求来获取,并不支持离线调试,使得脚本调试缺乏稳定性,而且由于脚本回放的速度受到 URL 请求发送和接收响应速度的影响而减慢,使得脚本调试的时间成倍增加。

针对在线回放反映的问题,我们设计了离线回放模式。离线回放是指回放浏览器根据 URL 请求,直接从本地 Web 资源文件中读取 Web 页内容,本地 Web 资源文件是在脚本录制时将 Web 页内容保存在本地而得到的。离线回放获取的网络资源可能不是最新的,但能够加快回放速度,而且不需要连通网络,实现本地调试。离线回放 Web 页内容显示速度取决于脚本回放所用机器的 CPU 和磁盘 I/O 速度,不受网络环境和 Web 服务器响应速度的影响。实现离线回放的关键技术是对 Web 资源进行本地存储。

2.3 Web 资源的本地存储之设计与实现

Web 资源的本地存储是指软件代理网关在录制时将接收到的 Web 资源,包括 Web 页内容、客户端头部、服务器端头部等,按照一定格式保存到本地资源文件中。在脚本修改调试回放过程中,可以直接从本地资源文件中读取并且显示脚本语句相对应的 Web 资源,如网页信息(htm、html、jsp 等)、图片信息(gif、jpg 等)、文本信息(txt、css、xml 等)、客户端头部服务器端头部等。将 Web 资源的本地存储用于虚拟用户脚本回放,可以使脚本调试内容保持相对的稳定,而不必每次显示脚本语句相对应的资源均要重新连接 Web 服务器来获取,进而加快回放速度,更重要的是允许进行离线调试。

在代理网关服务器上,为了加速 Web 资源访问速度,通常要对客户端经常访问的 Web 资源进行 Cache 缓存。虽然 Web 资源本地存储和代理网关的网络资源缓存均实现了 Web 资源的存储,但是代理网关的 Web 资源存储是以分散的 Web 页为基本单位的,而 Web 资源的本地存储则是将客户端发出的所有 URL 请求对应的 Web 页全部保存在一个结构化文件(.all)中,其结构更为紧凑,管理更为方便。

我们采用.all、.idx、.body、.chdr、.shdr 这 5 种文件格式来实现对 Web 资源的本地存储。其中.body 文件保存了 Web 资源的内容,.chdr 文件保存了获取 Web 资源时的客户端头部信息,.shdr 文件保存了返回 Web 资源时的服务器头部信息,.idx 文件则作为 URL 请求对应 Web 资源、客户端头部信息、服务器头部信息的索引文件,而.all 文件则包含了以上 4 种文件。在对录制脚本进行调试时,脚本中每一个 URL 请求均可以通过对.all 文件中保存的 Web 资源、客户端头部信息、服务器头部信息进行定位查找和显示,帮助测试人员分析脚本录制时的 Web 内容以及客户端、服务器头部信息。

设计实现 URL 资源的本地存储的关键是设计.all、.idx、.body、.chdr、.shdr 文件的内部结构和关系以及在.all 文件中对资源进行定位显示。5 种文件格式的结构设计如下:

(1).all 文件的内部结构:

.all 文件 = LastConnId(4 字节 Int) + BodyBase(4 字节 Int) + ClientHeaderBase(4 字节 Int) + ServerHeaderBase(4

字节 Int) + .idx 文件 + .body 文件 + .chdr 文件 + .shdr 文件

其中 LastConnId(4 字节 Int) + BodyBase(4 字节 Int) + ClientHeaderBase(4 字节 Int) + ServerHeaderBase(4 字节 Int) + .idx 文件合称为 Index 部分;.body 文件称为 Body 部分;.chdr 文件称为 ClientHeader 部分;.shdr 文件称为 ServerHeader 部分。LastConnId 表示本 Capture(.all)文件使用的最后连接 id 编号;BodyBase 表示 Body 部分在.all 文件中的开始位置;ClientHeaderBase 表示 ClientHeader 部分在.all 文件中的开始位置;ServerHeaderBase 表示 ServerHeader 部分在.all 文件中的开始位置。

(2).idx 文件的内部结构:

.idx 文件 = URL 请求字符串 + "\0" + Connid + BodyOffset + ClientHeaderOffset + ServerHeaderOffset

URL 请求字符串 + "\0" + Connid + BodyOffset + ClientHeaderOffset + ServerHeaderOffset

...

URL 请求字符串 + "\0" + Connid + BodyOffset + ClientHeaderOffset + ServerHeaderOffset

其中,Connid(4 字节 Int)表示连接 id 号;BodyOffset(4 字节 Int)表示该 URL 请求在 Body 部分中相对于 BodyBase 的偏移量;ClientHeaderOffset(4 字节 Int)表示该 URL 请求在 ClientHeader 部分中相对于 ClientHeaderBase 的偏移量;ServerHeaderOffset(4 字节 Int)表示该 URL 请求在 ServerHeader 部分中相对于 ServerHeaderBase 的偏移量。

(3).body 文件的内部结构:

.body 文件 = BodySize + QueryBody

...

BodySize + QueryBody

其中 BodySize(4 字节 Int)表示 Body 的大小;QueryBody 表示 Body 的内容,以"\0"结尾。

(4).chdr 文件的内部结构:

.chdr 文件 = ClientHeaderSize + QueryClientHeader

ClientHeaderSize + QueryClientHeader

...

ClientHeaderSize + QueryClientHeader

其中 ClientHeaderSize(4 字节 Int)表示 ClientHeader 的大小;QueryClientHeader 表示 ClientHeader 的内容,以"\0"结尾。

(5).shdr 文件的内部结构:

.shdr 文件 = ServerHeaderSize + QueryClientHeader

ServerHeaderSize + QueryClientHeader

...

ServerHeaderSize + QueryClientHeader

其中 ServerHeaderSize(4 字节 Int)表示 ServerHeader 的大小;QueryServerHeader 表示 ServerHeader 的内容,以"\0"结尾。

2.4 回放浏览器的设计与实现

我们在对回放浏览器实现机制进行研究的基础上,设计并实现了支持在线和离线两种回放模式的脚本回放浏览器 BView,用于对虚拟用户脚本进行回放以及回放结果的图像化显示。

BView 包括回放执行器和回放显示界面两部分。

回放模式通过菜单【选项】-【在线回放】/【离线回放】由测试人员选择。

回放执行器根据回放模式,选择是直接与 Web 服务器建

立连接,发送、接收、显示 Web 页内容;还是从 all 文件中直接读取本地 Web 内容,然后显示在回放显示界面中。

在线回放模式下,回放执行器根据 Connid 值与 Web 服务器建立编号为 Connid 的套接字连接,并且发送相关 URL 请求,接收到响应 Web 页后在回放显示界面中显示。离线回放模式下,回放执行器对一个 URL 请求,回放浏览器在 .idx 文件中查找到其对应的 BodyOffset 值,然后根据 BodyOffset 值在 .body 文件中获取 BodySize 和 QueryBody 值,并且将 QueryBody 内容作为 Web 页内容在回放显示界面中显示。在线和离线回放模式下,均需要将 Web 页以文件形式保存在本地目录中,并通过消息将 Web 页存放地址发送给回放显示界面进行显示。

回放显示界面由 URL 地址栏、回放控制按钮以及 Web 显示界面 3 部分组成。URL 地址栏显示当前 Web 页对应的 URL 地址;回放控制按钮包括最前、前一个、后一个、最后 4 个控制按钮,点击按钮可以前后浏览回放显示出的 Web 页;Web 显示界面用于显示在线和离线模式下的 Web 页,并且支持显示框架集,见图 1。

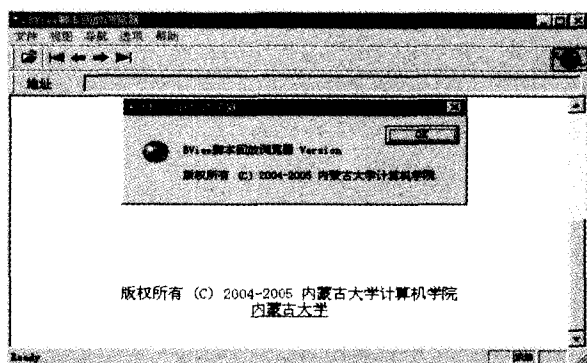


图 1 回放显示界面

2.5 回放浏览器实现验证

回放浏览器能否正确实现,关系到脚本是否符合业务逻辑和对 Web 服务器产生真正的压力,所以需要进行有效性验证。我们设计了两种测试方法,结合起来对 BView 进行验证测试,目的是通过结果来验证回放理论的正确性和实现的有效性。

方法一,在 Web 服务器端对 HTTP/S 请求日志进行记录,对比 IE 浏览器执行 HTTP/S 请求操作和 BView 回放脚本执行 HTTP/S 请求操作在服务器端产生的效果。

经过对日志文件的比较发现,从服务器端的角度看,BView 的 HTTP/S 请求操作和 IE 浏览器的 HTTP/S 请求操作内容和效果完全相同,说明 BView 回放浏览器的实现是正确的。

方法二,使用 Sniffer 软件工具,对网络数据包进行监听,检查 IE 浏览器和 BView 发包内容的一致性。

经过对 Sniffer 软件监听结果文件的比较发现,除了 sessionid 和 cookie 的内容每次测试的内容不同,IE 浏览器和 BView 发包的其余内容一致。在 HTTP/S 协议下,每次客户端与服务器端建立连接时,服务器端均会为客户端产生新的 sessionid 和 cookie,所以会产生发包内容的个别差别,这是正常的。尤其是,如果 sessionid 每次不发生改变,会导致服务器端因为客户端 sessionid 与服务器端 sessionid 不一致而拒绝客户端请求的现象发生,从这一方面来说,sessionid 的内容

的变化是必要的。从网络数据包的监听结果来看,也说明 BView 回放浏览器的实现是正确的。

3 Web 回放模型

Web 回放浏览器的评价,需要从速度、功能等方面综合考虑。我们给出了 Web 回放模型 WRM、Web 回放模型要素优先级以及 Web 回放效果 WRE,用以帮助测试人员对 Web 回放浏览器进行定量的分析和评价,找出其优劣之处,进而促进 Web 回放浏览器的研发工作。

3.1 Web 回放模型定义

定义 1 Web 回放模型(Web Replay Model, WRM)是一个四元组 (S_i, M_i, P_i, L_i) ,其中 S (Speed)表示回放显示速度, M (Mode)表示回放模式, P (Page)表示回放对页面的显示, L (Logic)表示回放是否符合业务逻辑, i 表示相关该类的要素编号。

S 、 M 、 P 、 L 四类要素中的子要素,可以根据实际情况进行增减修改,从而修改模型来满足不断变化的 Web 回放。这里,我们对模型进行了相关如下定义。 S 包括两个子要素: S_1 表示对 HTTP 主请求的回放显示速度, S_2 表示对 HTTP 从请求回放显示速度; M 包括两个子要素: M_1 表示是否支持在线回放模式, M_2 表示是否支持离线回放模式; P 包括 3 个子要素: P_1 表示支持静态页面回放, P_2 表示支持动态页面回放, P_3 表示支持框架(Frame)显示; L 包括两个子要素: L_1 表示静态业务逻辑, L_2 表示动态业务逻辑。

定义 2 Web 回放模型要素优先级,为 S 、 M 、 P 、 L 四类要素之间以及同类要素之间在 WRM 中所对应权重。权重高表示优先级高,反之优先级低。权重可以根据实际情况进行自定义,一般为正整数,0 表示最低权重。要素优先级按要素类由高到低顺序为, $L > P > S > M$; 按同类要素排序为, $L_1 > L_2$, $P_1 > P_2 > P_3$, $S_1 > S_2$, $M_1 > M_2$ 。

定义 3 Web 回放效果(Web Replay Effect, WRE)定量地表示 Web 回放浏览器的回放效果。WRE 值定义为 WRM 中 S 、 M 、 P 、 L 四类要素中各子要素加权值算术和,表示为:

$$WRE = \sum_{i=1}^{n_1} w_s S_i + \sum_{i=1}^{n_2} w_m M_i + \sum_{i=1}^{n_3} w_p P_i + \sum_{i=1}^{n_4} w_l L_i$$

其中 n_1 、 n_2 、 n_3 、 n_4 分别表示 S 、 M 、 P 、 L 四类要素中子要素的个数, w_s 、 w_m 、 w_p 、 w_l 分别表示 S 、 M 、 P 、 L 四类要素对应的权重。WRE 值越大,表示 Web 回放浏览器的回放效果越好,最小值为 0。

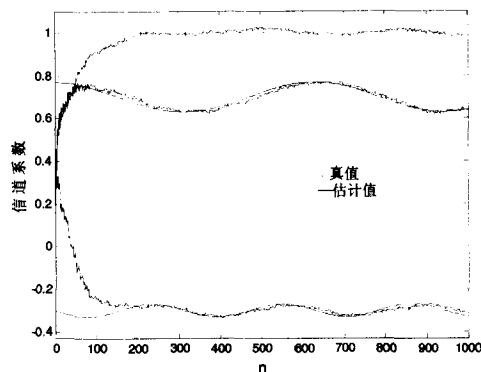
WRE 可以进行自定义,主要表现在各类要素的权重以及各自要素的值可以根据一定的标准进行自定义。

3.2 BView 与 Run-Time Viewer 功能比较

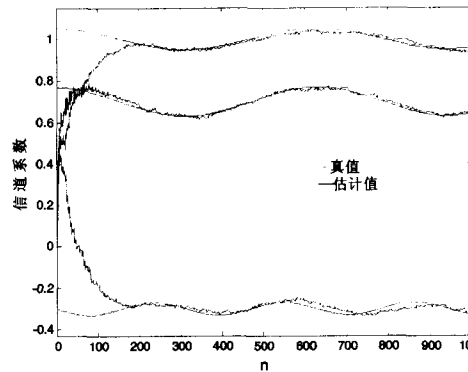
LoadRunner 作为当前 Web 性能测试工具的佼佼者,自身也带有一个回放浏览器 Run-Time Viewer,用于显示脚本回放内容。Run-Time Viewer 在回放脚本时,首先进行语法编译检查,读取脚本中的 URL 地址,并且连接 Web 服务器获取最新 Web 资源进行显示。Run-Time Viewer 的功能与 BView 的在线回放模式相同。但是 Run-Time Viewer 不支持离线回放模式以及框架集的显示,其在显示带框架的 Web 页时,并不是将组成框架集的所有框架显示在一个 Web 页中,而是将每个框架作为一个独立的 Web 页进行显示,这种处理方式破坏了框架集的整体性。

相比之下,BView 的在线、离线两种任意选择的回放模

(下转第 65 页)



(a) 时变信道模型 I



(b) 时变信道模型 II

图 8 多径时变衰落信道的盲均衡实现(SNR 为 30dB, 标准的 LMS 算法被采用)

结论 本文将一种非线性滤波与神经处理单元相结合成了一种新的滤波算法,它有解决状态空间模型的估计问题的能力。在将基于混沌的通信系统处于多径衰落信道的通信场景时的盲信道均衡问题转化为增广的状态空间模型以后,该滤波算法成功地解决了混沌通信系统经历多径衰落环境的盲信道均衡问题,与标准的 LMS 算法相比,该算法具有较快的收敛速度和较好的滤波性能。计算机仿真证实了有关结论。

参考文献

1 Lau F C M , Tse C K. Chaos-based Digital Communication Sys-

tems. Heidelberg, Germany: Springer-Verlag, 2003
 2 Feng Jiuchao, Tse Chi Kong . Reconstruction of Chaotic Signal with Applications to Chaos-based Communications. 清华大学出版社.(新加坡)世界科学出版公司,2005
 3 Iltis R A, Fuxjaeger A W. A Digital DS Spread-Spectrum Receiver with Joint Channel and Dopple Shift Estimation. IEEE Transactions on Communication, 1991,39:1255~1267
 4 Howard S I, Pahlavan K. Autoregressive modeling of wide-band indoor radio propagation. IEEE Trans. on Commun. , 1992, 40: 1540~1552
 5 Haykin S. Adaptive Filter Theory. Fourth Edition. New Jersey: Prentice Hall, 2001
 6 阎平凡,黄端旭. 人工神经网络—模型·分析与应用. 安徽教育出版社,1993

(上接第 60 页)

式,以及支持对框架集的显示,则能更好地满足测试实践中测试人员的要求。

3.3 BView 与 Run-Time Viewer 的 Web 回放效果比较

我们采用 Web 回放效果值对 Run-Time Viewer 回放浏览器和 BView 回放浏览器进行了定量比较。

我们在进行定量比较时,将要素权重分为 1、2、3、4 共 4 个递增级别,1 表示不重要,2 表示一般重要,3 表示比较重要,4 表示非常重要。根据 Web 回放模型要素优先级,将 w_s 、 w_M 、 w_P 、 w_L 值分别设定为 1、2、3、4,而 n_1 、 n_2 、 n_3 、 n_4 值分别为 2、2、3、2。代入 WRE 公式,得

$$WRE = \sum_{i=1}^2 S_i + \sum_{i=1}^2 2M_i + \sum_{i=1}^3 3P_i + \sum_{i=1}^2 4L_i$$

我们对 WRM 中所有要素进行评分,评分标准分为 0、1、2 共 3 个递增级别,0 表示不好或不支持,1 表示一般,2 表示好或支持。BView 回放浏览器采用支持在线和离线回放模式和框架显示,但对动态业务逻辑显示一般。Run-Time Viewer 回放浏览器不支持离线回放模式和框架显示,但在动态业务逻辑方面效果较好。两种回放浏览器的 WRE 值分别为:

$$WRE_{BView} = 1 * (2+2) + 2 * (2+2) + 3 * (2+2+2) + 4 * (2+1) = 42$$

$$WRE_{Run-Time Viewer} = 1 * (2+2) + 2 * (2+0) + 3 * (2+2+0) + 4 * (2+2) = 36$$

表明在 Web 回放效果方面,BView 回放浏览器相对 Run-Time Viewer 回放浏览器总体占优。

总结与进一步工作 我们在第 1 节分析了回放浏览器在 Web 性能测试中的作用。第 2 节提出了支持在线和离线两

种回放模式的 Web 性能测试回放浏览器 BView 以及离线回放实现中的关键技术——Web 资源本地存储的设计与实现,并且对 BView 的实现进行了验证。第 3 节给出了一个 Web 回放模型,在 BView 同 Run-Time Viewer 进行功能对比的基础上,指出 Run-Time Viewer 的不足是不支持离线回放和框架集的显示,并且采用 Web 回放效果 WRE 对 BView 和 Run-Time Viewer 进行了定量比较。

BView 回放浏览器对 WRM 中所有要素均支持,但在一些方面仍然有待改善。它显示 HTTP 主请求和从请求速度一般,主要因为其需要许多磁盘 I/O 进行 Web 页读写操作。可以考虑在内存中开辟 Web 页文件空间,一次性将 all 文件读入内存,在内存文件中实现读写操作,从而加快显示速度。而为了使 BView 在动态业务逻辑方面提供更好支持,还需要对相关机制进行深入的研究。

参考文献

1 武海平,蒋东兴,等. Web 服务器通用性能测试系统的设计与实现[J]. 小型微型计算机系统,2003,24(2):188~190
 2 许蕾,徐宝文. 用户行为获取方法在 Web 性能测试中的应用研究[J]. 软件学报,2003,14(增刊):115~120
 3 刘彤,周工业,倪聚铭. HTML 浏览器的设计与分析[J]. 计算机工程与应用,2001,37(9):117~119
 4 许蕾,徐宝文,陈振强. Web 测试综述[J]. 计算机科学,2003,30(3):100~104
 5 阳富民,周艳,周正勇. WML 浏览器的设计与实现[J]. 计算机工程与科学,2004,26(9):4~6
 6 王少春,陈家骏,王启祥,等. Internet 在线翻译浏览器技术探讨[J]. 计算机应用研究,2001(1):11~13,36
 7 戎晓霞,王金栋,吴胜远. 基于 BHO 和协同技术的多级文语的实现[J]. 计算机工程,2004,30(2):42~44
 8 梁哲炜,宋小波,等. 一款新型盲用浏览器的设计与实现[J]. 计算机工程与应用,2004(14):106~108