

基于微型字体服务器的多语言 HTML 文本显示方法

党磊¹, 党德玉²

(1. 英国曼切斯特大学 计算机科学系 Manchester M13 9PL UK; 2. 东北电力学院 信息工程系 吉林 吉林 132012)

摘要: 介绍了一种实现多语言 HTML 文本显示的方法, 该方法通过建立微型字体服务器, 实现字符信息的非图像传输, 节约了传输带宽, 提高了传输速度。讨论了该方法的基本原理, 实现的方法, 与传统方法相比较, 阐述了该方法的特点。

关键词: 多语言显示; HTML; Java 文本显示

中图法分类号: TP311.11

文献标识码: A

文章编号: 1001-3695(2004)09-0217-02

An Approach to Multilingual HTML Text Displaying

DANG Lei¹, DANG De-yu²

(1. Dept. of Computer Science, University of Manchester Manchester M13 9PL UK; 2. Dept. of Information Engineering, Northeast China College of Electric Power Engineering, Jilin Jilin 132012, China)

Abstract: This paper introduces an approach to multilingual HTML text displaying. By means of a tiny font server, the characters are transmitted in a non-image form between Web server and client terminal, which can save bandwidth of transmission, improve the transmitting speed and etc. The principle, realizing process are discussed. Compared with traditional one, the characteristic of this new approach is also described.

Key words: Multilingual Displaying; HTML; Java Text Displaying

随着互联网技术的发展, Web 服务作为 Internet 上最大的一种信息源已经深入我们生活中的各个方面。在应用的过程中出现的主要问题之一, 就是不同语言字符集的显示问题。

为解决多语言显示问题, 无平台依赖性的 HTML 文本显示方法逐渐浮现出来。该技术的核心在于解决如何在不需要用户平台支持的情况下正常显示未包含于 ISO-8859.1 字符集中的字符文本。这一技术看似简单, 实则复杂, 如果客户端只能支持通用的 HTML 浏览器以及其他可以在浏览器环境中运行的程序语言, 并且不能通过在客户端安装其他附加应用程序或库文件的方式来获得相关语言支持, 文本显示则必须由一个多语言 HTML 系统来支持, 而无法依赖用户的操作系统或能够支持该语言的浏览器。

本文提出了一种基于微型字体服务器的多语言文本显示方法, 该方法克服了传统方法中的一些缺点, 如要求的传输带宽较宽, 传输信息量较大, 传输速度较慢等。

1 多语言编码与显示原理

在浏览器中显示字符, 如果没有操作系统或者浏览器的该语言支持, 字符从编码到图像的转换过程就必须由相应的显示支持系统来完成。而后再将这些字符对应的图片按照既定的格式组织起来。通常, 图片的产生过程可以分为静态产生, 即预先将所有可能用到的字符(甚至是整个字库)的相应图片都存储在服务器上备用, 另一种方法则是动态产生, 即当客户端请求某一特定字符显示时候再为其产生相应的图片文件。显

然, 这两种办法的优缺点明显: 前者响应速度快, 但是占用大量的资源; 后者响应速度慢, 但除字库外不需要占用额外的存储空间。两种方法都有一个共同的优点: 如果客户浏览器具有缓存机制, 那么当绝大多数常用字符都已经存在于缓存中时, 则由于省去了请求和传输字符图片的时间, 文本的显示速度将大大加快。由于它们传输的都是字符的图形信息, 所以它们也有一些共性的问题: (1) 占用较大的传输带宽。一张 16 × 16 点阵的黑白图片, 即便使用 GIF 格式存储, 也需占用 178 字节的存储空间, 如果是几千、几万不重复字符同时传输, 那么将占用相当可观的带宽。(2) 字符图像的形式相对固定, 字符的一些基本特性, 如旋转、颜色等难以表述, 如果为这些属性及其组合单独准备图片, 显然是不现实的。(3) 用于图片的大量 标记也增大了 HTML 文件的长度。

2 网关服务程序的解决方案

目前研究较多的解决方法是基于网关服务程序的解决方法。方法的原理如图 1 所示。

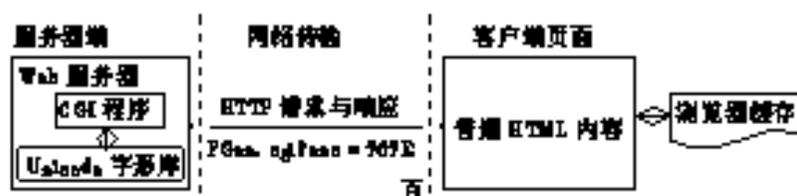


图 1 CGI 方法结构示意图

基于网关服务程序(Common Gateway Interface, CGI)的解决方案本质上是传统方法的一种改进。该方法动态生成所需字符的图像文件, 并给予图片文件一定的自由度, 如颜色甚至大小。这一方法在客户端无特殊的资源等需求, 而在服务器端

需添加一个负责字符文件生成的 CGI。使用具有绘图能力的 CGI 脚本语言,如 Perl, PHP, JSP 都可以完成这一功能。

字符显示所需要的图片由服务器端的 CGI 程序来生成。Web 服务器传送给客户端的 HTML 源代码中,需要显示的非 ISO-8859 字符的位置都由图片替代,CGI 程序查询字形库并生成相应的图片文件供浏览器显示。还可以为请求附加其他参数,如颜色等,用于指定当前字符的颜色。

该方法在速度上很大程度依赖于客户端的浏览器缓存。如果浏览器采用合理的缓存机制,在一定时间间隔之后,大部分常用的字符图片便已经处于浏览器的本地缓存中,接下来显示过程中,由于大部分常用字都不需要再次重复请求,显示速度能够得到较大的提升。

该方法的主要弱点是耗费较大的带宽,每一字符都需要 178 字节,同时对浏览器有很强的依赖性,浏览器的缓存机制将决定显示的效率。

当服务器的负载较重,而且面对频繁的请求时,CGI 程序的执行将占用大量的系统资源。因此,该方法未摆脱常规方法的约束,具有传统方法的基本特征。在传统方法中,使用自行编写的 Web 服务器可能是较为行之有效的提高效率的方法,如在 UNIX 系统中,可以使用系统所提供的 Select() 函数编写一个简单的 HTTP 服务器,以此来提高系统运行效率。基于上述思想的解决方案,由于仍然采用的是图像信息的远距离传输,对有速度、带宽约束等系统的应用,具有明显的弊端。

3 微型字体服务器的解决方案

3.1 基本原理

本方法与上述方法的基本区别在于通过建立微型字体服务器,实现字符的字形信息传输,而不传输字符的图像信息。方法的核心是提供字形信息的微型字体服务器(Tiny Font Server)。而客户端则负责为 HTML 提供必要的多语言字符显示支持。如可使用 Java 编写这一客户端程序,将其内嵌至 Web 页面内运行。Java 客户端负责向 TFS 请求字形信息,并对获得的字形信息进行相应的还原处理。

由于字符可以表示为 16 ×16 的点阵,本方法将这一点阵以数组的方式自 TFS 传送给客户端,考虑到所要显示的字符几乎不可能出现同一字符内有多种颜色的情况,那么实际上这一点阵所需要的数据量为 16 ×16 ×1 位 = 32 字节。这种情况下,请求信息若采用 UTF-8 编码,平均传输数据量为 2 字节,而返回的数据仅需 32 字节,则每字符的数据量仅为锐减到了 34 字节。

为实现上述思想,在服务器端需设置字体服务器,用来接收来自客户机端的请求,在处理该请求后传输相应字形信息,并在客户端设置字符显示模块来发出请求和提供显示服务。在这种情况下,每个字符所需要传输的数据量最多仅为 36 字节。与前述的 178 字节相比较,传输的数据量有了巨大的改善。

3.2 系统结构及其实现

TFS 方法的基本结构如图 2 所示。

由图 2 可知,TFS 方法中重要的部分是建立在服务器端的微型字体服务器。TFS 和系统原有的 Web 服务器是两个分离的模块,Web 服务器部分可以使用常规流行的服务软件,如

Microsoft IIS 或者 Apache,亦可自行编写简单的 Web 服务器程序。这种情况下,虽然为其加入 CGI 支持较为困难,但自行优化的 Web 服务程序将极大地提高系统性能和简便程度。Web 服务器通过 HTTP 协议(通常运行于 80 端口)与客户端的浏览器进行通信,接收请求并发送相应的 HTML 页面源代码及其他资源。

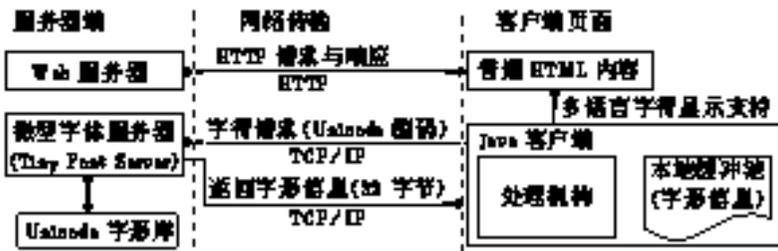


图 2 TFS 方法结构示意图

TFS 作为系统的一个后台程序,以服务或者守护进程的形式运行于系统中,负责侦听一个指定的端口上的连接请求。当 TFS 接收到客户端发送来的字符请求后,查询 Unicode 字形库得到这一字符所对应字形信息,将该字符的点阵表示按 1 位/点的规则转换为 32 个字节的字符串后,再将这一字符串返回给客户端。

客户端程序运行于浏览器中,并由浏览器负责 HTML 页面以及其他资源的显示。在需要使用到非 ISO-8859 所标定的字符时,由运行于客户端中的 Java 程序检查该字符是否已经存放于本地缓存中,若本地缓存中没有相应字符,则向 TFS 发出请求,请求内容为该字符的 Unicode 编码,并接收来自 TFS 的返回串。如本地缓存中已有该字符,则直接从本地缓存中取得相应的字符信息。而后 Java 程序将这一 32 字节的字符串还原成 16 ×16 的点阵图像,并将这一图像显示在相应的位置,完成字符显示全过程。

Java 客户端内的本地缓冲池可用于存放近期使用过的字形信息,目的在于减少重复请求次数,节约网络传输时间来提高显示的速度。本地缓冲池可以是链式结构,以字符的 Unicode 编码作为关键字,考虑到关键字数目较多,可将关键字进行 Hash 之后链入相应的链,并按照一定的时间约束策略对长时间未用到过的记录进行淘汰。

本方法的原理工作过程描述如下:

首先定义两个操作

(1) 操作 A: Java 客户端将指定字符的 Unicode 编码作为请求发送给 TFS; TFS 接收请求,查找字形库得到该字符的 16 ×16 点阵表示; TFS 按照每点 1 位的方法将 16 ×16 点阵转换为 32 字节的串; TFS 将该 32 字节返回给 Java 客户端; Java 客户端接收该串,并将其存入本地缓冲池。

(2) 操作 B: Java 客户端将 32 字节的串还原为背景透明的位图; 显示指定字符。

字符显示过程如下:

(1) Java 客户端接收到显示某个字符的请求;
(2) Java 客户端首先查找本地缓存,确认该字符是否已经存在: 找到,从本地缓存中读取对应的 32 字节; 未找到,执行操作 A。

(3) 执行操作 B。如以汉字“百”字为例,Java 客户端发出的请求为 767E(“百”的 Unicode 编码),而 TFS 所返回的信息为(十六进制表示)00 00 FF FE 03 00 02 00 1F F0 10 10 10 10 10 10 1F F0 10 10 10 10 10 10 10 10 10 10 1F F0 10 10,而后客户端依据这一串字符将其还原为点阵表示。(下转第 221 页)

统平台和其相应的协商 Agent 进行交互; 协商 Agent 接收选择代理的 XML 文档, 转换成 ACL 消息; 双方按照一定的策略讨价还价, 达成成交条件并形成合同, 合同用 XML 语言描述; 买方的协商 Agent 带着合同返回。

(5) 双方进行交易: 购买商系统的交易 Agent 携带用户的付款信息、认购、数字签名、货币等私人信息, 与电子银行、认证中心等协同工作, 遵循安全交易协议, 完成身份确认, 资金转账等工作, 然后供应商交付商品。

(6) 交易后的处理措施: 完成交易后将这次交易的信息存到知识库中; 由个人助理 Agent 根据知识库中的信息和这次交易的一些信息得出一些商家所关心的数据存储到相应的数据库中, 并且评价这次交易。另外供应商系统的售后服务 Agent 负责商品的一些售后服务。

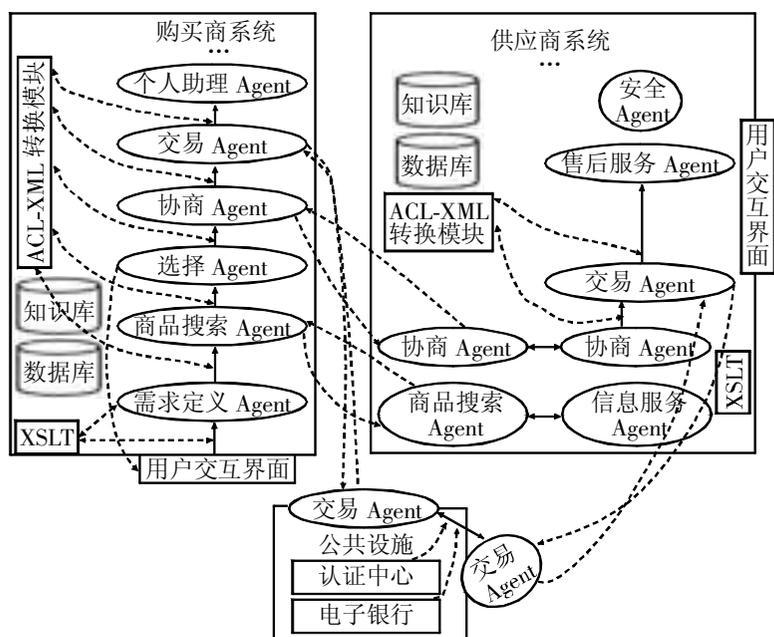


图 2 模型工作流程图

4 结束语

Mobile Agent 技术在电子商务模式中具有很多优越性, 它不但为商务主体提供了迅速接入 Internet 搜索自己所需商品的智能技术, 而且免除了交易双方在交易中为建立、进行和完成交易所耗费的大量时间和精力。另外, 使用 ACL-XML 作为 Mobile Agent 的通信语言进行电子商务中的数据交换有着其他语言所无法比拟的优点。这两种技术在电子商务的发展中有着很好的前景。

本文结合这两种技术构建了一个采用 ACL-XML 作为通信语言的 Mobile Agent 的电子商务模型, 讨论了此模型的工作流程。但是对于此模型中的其他一些问题, 如安全机制, 容错机制等还有待进一步讨论研究。我们坚信, 这两种技术本身的不断成熟和在电子商务的应用的不断发展, 一定会使得电子商务技术遍地开花。

参考文献:

- [1] 陈斌亮, 陈鸿, 等. 基于移动代理的电子商务框架[J]. 计算机工程, 2001, 8(27): 160-161.
- [2] 陶先平, 吕建, 李新, 等. 移动 Agent 技术在电子商务上的应用初探[J]. 南京大学学报, 2001, 2(37): 174-182.
- [3] 谢用辉, 张宝行, 等. 一个网上数据交换的新技术——XML 的分析和实现[J]. 计算机工程与应用, 2002, (5): 153-155.
- [4] IPA. FIPA ACL Message Structure Specification[EB/OL]. <http://www.fipa.org/specs/fipa00061>, 2001.

作者简介:

郑晓梅(1980-), 女, 江苏兴化人, 硕士研究生, 主要研究方向为电子商务, J2EE 开发; 张天, 硕士研究生, 主要研究方向为软件工程——MDA、J2EE 开发; 夏阳, 副教授, 硕士生导师, 主要研究方向为电子商务。

(上接第 218 页)

3.3 方法评价

本方法的特点主要有: (1) 节约带宽。每个字符只需不足 40 字节的数据交换。(2) 支持颜色。在生成字符时, 可以按照指定的颜色方便地生成单色字符。(3) 显示速度快。本地缓存可以非常明显地提高重复字符的显示速度。(4) 多平台特性好。Java 所编写的客户端程序可以在任何支持 Java 语言的平台上运行, 如果使用 Active X 组件来实现将更加方便。

这一方法的缺点主要有: (1) 需要浏览器支持 Java 运行环境。目前流行的浏览器已经全部支持 Java。(2) 缓存初始化。由于 Java 的生存周期是每个页面, 故因为每次页面重载时都会导致重新初始化缓存, 所以页的显示开始阶段较慢。(3) 可能不能运行于多重防火墙后。在 Java 的默认安全模式下, 客户程序仅能同其相关的服务器进行通信, 那么如果用户所用计算机处于多个防火墙后, 则此方法将无法运行。但可将数据封装在 HTTP 协议数据中进行传递来加以解决。

测试结果表明, 本方法的传输量仅是传统方法的二分之一。

4 结论

本文涉及的两种方法有其各自的应用环境要求。在条件限制比较严格的情况下, CGI 方式将是较好的选择, 因为其不

受浏览器以及网络运行条件的限制, 但要求较大的网络传输数据量。在绝大多数情况下, 由于系统可提供了实现的环境, TFS 方法将能够得到非常好的应用效果, 如传输速度快, 传输时间短, 等等, 实现也不复杂。因此, 本文提出的方法(TFS)有着广阔的应用空间, 优势也是明显的。TFS 方法和 CGI 方法的分别强调了问题的不同的侧面, 有着各自的特点, 两种方法结构上不尽相同, 在传输的内容上有着本质的区别。对 TFS 方法也正在优化和改进之中, 相信 TFS 方法会是更易于使用, 更有前途的多语言 HTML 文本显示方法。

参考文献:

- [1] The Unicode Incorporated, The Unicode Standard: A Technical Introduction[EB/OL]. <http://www.unicode.org>, 2003.
- [2] Rohit Khare, et al. Composing Active Proxies to Extend the Web[J]. Workshop on Compositional Software Architectures, 1998.
- [3] Shigeo Sugimoto. Experimental Studies on an Applet-based Document Viewer for Multilingual WWW Documents- Functional Extension of and Lessons Learned from Multilingual HTML[C]. Proceedings of the Second European Conference, ECDL '98, Heraklion, Crete, Greece, 1998. 375-378.

作者简介:

党磊(1981-), 男, 博士研究生, 研究方向为人工智能技术及其应用、大型数据库技术; 党德玉(1950-), 男, 教授, 系主任, 主要研究方向为人工智能技术及其应用。