

基于内容协商实时在线翻译方案的实现及分析

王 仆¹, 钟宝燕², 李 翔¹, 李建华¹

(1. 上海交通大学 信息安全工程学院, 上海 200240;

2. 上海政法学院, 上海 201701)

摘 要:互联网上大量的信息往往以不同种的语言出现,为了在尽可能短的时间内了解这些信息,也为了在互联网舆情管控领域的中文文本处理中使用这些信息文本,需要借助于在线实时网页翻译。文中在分析了传统网页翻译方法的不足后,提出了基于内容协商和网络缓存的网页实时在线翻译系统的方案,使得翻译服务对于客户端透明,节省客户端多余操作时间,同时使得对于重复请求的网页呈现效率得以提高。并通过分析和实验证实了该方案相对于传统网页翻译方法的优越性。

关键词:内容协商;网页翻译;网络缓存

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2008)03-0009-04

Realization and Analysis for Resolutions of Web Page Online Realtime Translation Based on Content Negotiation

WANG Pu¹, ZHONG Bao-yan², LI Xiang¹, LI Jian-hua¹

(1. Department of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200240, China;

2. Shanghai University of Political Science and Law, Shanghai 201701, China)

Abstract: The large number of information on the Internet is always in different language. In order to understand these information and also in order to use these content of information in the area of Internet consensus management and control, have to use the online real-time translation. After analyzing the disadvantage of traditional way of Web page translation, gives the new resolution based on the mechanism of content negotiation and Web cache, which makes the translation service invisible to the client, saving the additional operating time. And also improves the efficiency of Web page presenting when repeating to request. Through experiments, proves the improvement of the new resolution by comparing with the traditional Web page translation method.

Key words: content negotiation; Web page translation; Web cache

0 引 言

国际互联网(Internet)的普及,使得上网成为了一件十分平常而又不可缺少的事情。为了得到最新的网络资料,很多人会选择通过搜索引擎来查找,但是往往会发现很多最新的资料通常是英文的,甚至是法文、日文等等,如碰到不认识或不熟悉的语种就难免望文生畏。

目前的解决方法是通过专业的翻译软件^[1],或者直接通过在线的第三方翻译服务提供者来进行全网页翻译^[2]。但是上述方法的效率其实都不高,首先翻译需要更多的额外操作来实现,用户需要通过浏览器登陆网页翻译服务页面,手动粘贴需翻译的目标网页的链接或翻译内容,然后选择具体要翻译的类型,这使得浏览网页资料无法高效地进行;其次在没有 Web Cache 支持下,反复向第三方翻译服务器请求网页翻译服务必将浪费大量网络带宽。

文中在首先说明网络客户端直接向第三方翻译服务器请求网页翻译服务——这一目前最为常见的网页在线翻译手段的基础上,重点介绍了基于代理服务器利用网络缓存(Web Cache)技术和 HTTP 协议所包含的内容协商机制,构建网页实时、在线翻译系统的具体方案。

收稿日期:2007-06-14

基金项目:上海市科委资助项目(065115020);国家自然科学基金资助项目(60502032);教育部新世纪优秀人才支持计划资助项目

作者简介:王 仆(1982-),男,上海人,硕士研究生,研究领域为计算机网络应用层,内容安全及内容过滤;李 翔,副教授,博士,研究方向为网络内容传输效率与安全管理、网络舆情监控与预警、网络多媒体对象内容分析等;李建华,教授,博导,研究方向为内容安全管理、电子政务、网络安全综合管理等。

1 背景知识

1.1 网络缓存技术

在通过代理服务器实现网页实时、在线翻译的过程中,客户端以 Web 浏览器形式出现,浏览器下载好目标网页源文件之后并不能够马上将其内容显示,而需要花一定的时间来解析源文件,下载源文件中的网页对象,运行潜在的本脚本片段。鉴于此,从得到实现翻译的源文件到浏览器显示客户端所需内容的时间,是判断实时网页翻译效率的重要指标之一。

在网页呈现过程中,浏览器通常需要下载一定数量的网页内嵌对象,这其中静态的图像文件所占比重相对较大,代理服务器对于这些图像文件通常是能够实现内容缓存的^[3]。因此,在网页实时、在线翻译系统中引入网络缓存技术,能够显著降低对于同一网页,甚至来自相同站点的类似网页重复请求时,浏览器所花费的显示网页时间,提高网络内容传输效率。

1.2 内容协商机制

HTTP/1.1 协议中定义了内容协商的工作模式,即客户端在 HTTP 请求包的头部中加入特殊字段来选择获取远程服务器上资源的不同版本^[4,5]。如果服务器上有符合客户端要求的资源版本时,服务器选择其中最为符合的内容返回给客户端。HTTP/1.1 协议定义了以服务器为驱动和以客户端为驱动的两种基本模式^[6]。在具体实现过程中,通常是把两种基本模式相结合,形成对客户端而言透明的内容协商机制^[7]。

在通过代理服务器实现网页实时、在线翻译的过程中,透明协商起到了关键的作用。代理服务器通过与网络客户端进行内容协商,在获知客户端对于网页翻译服务的具体要求后,利用第三方翻译服务器将客户端请求的目标网页翻译成内容协商过程中确定的语种予以返回。文中是通过在网络客户端数据请求过程中增加协商环节,实现网页实时、在线翻译系统中的内容协商机制,对客户端请求数据包的获得与解析工作由改进后的网络缓存技术完成。

2 实时在线网页翻译方案

在实时在线网页翻译的方案性能比较中,最主要的是从发出请求到所需页面中主要内容呈现完成的时间。同时假设翻译的质量在可接收的翻译效率下是可以得到保证的,并且翻译所需时间并不受到网络因素影响,只同提供服务的机器性能及相关程序有关。

为了方便论述,做出如下定义:

1) 网页呈现总时间(T):指从发出请求到目标页面中的主要内容呈现完成所需时间。为了效率考虑,网页中主要内容呈现后浏览器转入停止状态时,不需

要其余的网页冗余信息。所以该时间区别于网页完全呈现时间。

2) 源文件获取时间(FGT):指从客户端发出请求到其获得第三方翻译服务器返回的,已完成翻译的目标网页源文件所需时间。在翻译服务中,主要就是对网页源文件中的字符语言种类的转换,所以 FGT 的大小其实体现了翻译效率以及所处网络的传输效率。

3) 网页内容呈现时间(PST):指从获得目标网页源文件后到目标页面中主要内容呈现完成所需时间。由于语言种类转换在源文件中已经进行,所以 PST 同翻译的效率无关。由于网页的呈现主要取决于 PST, PST 总体上同浏览器的性能、目标站点采用的 Web 标准,以及所处网络的传输效率有关。而 PST 又同具体的某个页面中必须呈现的对象及其大小有关。

上述三个时间的关系为:

网页呈现总时间(T) = 源文件获取时间(FGT) + 网页内容呈现时间(PST)

实时在线网页翻译主要是从上述三个时间来判断其性能。

2.1 传统网页在线翻译方法(简称原始方案)

由客户端向第三方服务器直接请求,第三方服务器代为请求目标网页,再经过翻译后返回给客户端。这种方案即为一般最常见的方法,其中没有代理服务器的参与,并未引入内容协商和网络缓存技术。

原始方案在操作方面花费多余的人工时间。第三方翻译服务器只是提供网页文本翻译的服务,客户端必须进行相关操作,包括获得目标网页的 URL,登陆第三方翻译服务器进行手动输入,选择翻译的原语种与目标语种等等。

其次,原始方案在对于重复对象的请求上浪费不必要时间。客户端将请求发送给第三方翻译服务器之后,第三方翻译服务器将代为请求目标网页的源文件,之后将目标网页源文件进行翻译后传送给客户端。客户端接收到已实现翻译的网页源文件之后,由浏览器进行必要的处理,直接向目标站点获取网页上的内嵌对象以便呈现网页。由于并未引入网络缓存技术,对于请求的静态网页及其中的静态对象无法实现内容缓存。因此,尽管网页上的内嵌对象是没有必要进行翻译的,但是这些内容在反复请求过程中仍然需要占据大部分传输时间,降低网页翻译的整体效率。

另外,第三方翻译服务器的相对不稳定性会造成原始方案的不稳定性:一是可能会无法保证服务的稳定提供;二是第三方翻译服务器可能对于用户要求进行翻译的目标网页无法成功获取,导致翻译服务失败。

由分析可知,原始方案中,如果不考虑额外的人工

操作时间,仅分析从请求发出到页面主要内容呈现的时间,FGT 取决于第三方翻译服务器的翻译效率以及客户端到第三方翻译服务器、第三方翻译服务器到目标站点的网络延迟。而 PST 则基本与第三方翻译服务器无关,只同浏览器性能及客户端同目标网站间的延迟有关。

2.2 基于内容协商和网络缓存网页在线实时翻译(简称为改进方案)

增加代理服务器,引入内容协商和网络缓存技术,在客户端同代理服务器进行内容协商后,由代理服务器完成对客户端目标网页 URL 的解析工作,并向第三方翻译服务器发送请求。在网页实时、在线翻译系统中,第三方翻译服务器还是起到代为请求目标网页,并且提供网页翻译服务的作用。网页实时、在线翻译系统中代理服务器的功能流程如图 1 所示,图中灰色部分说明在传统的內容缓存模块中增加内容协商机制的具体实现办法。

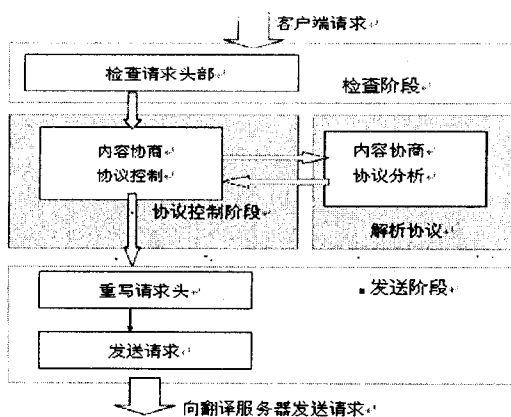


图 1 代理服务器功能流程图

考虑到使用网络浏览器的客户端通常不易修改请求数据包中的内容,因此可在 HTTP 协议的基础上,对于客户端发送的 URL 进行修改,达到内容协商的目的。文中使用的方法是,在客户端所请求的目标网页 URL 后添加字符串“- - ConNego - - origin = en&target = zh”,表明目标网页使用的是英文,而客户端要求将其翻译成中文。对客户端目标网页 URL 的解析工作由网络缓存中的内容协商模块完成,代理服务器将实现解析的目标 URL 发送给第三方翻译服务器。第三方翻译服务器仍然进行代为请求与内容翻译操作,对其返回的、已实现内容翻译的数据,代理服务器在完成内容缓存的同时向对应的请求客户端响应。

改进方案中的客户端同代理服务器之间通过内容协商功能来实现内容自适应的机制,对于客户端的请求由代理服务器处理并同第三方翻译服务器交互,获得客户端所需要的经翻译的网页。这对于客户端是透

明的,基本避免了人工操作所花费不必要的时间。

代理服务器的缓存机制对于可以缓存的网页对象的缓存使得重复请求不需要再次通过向目标服务器请求,而由代理服务器直接返回给客户端,从而使得重复请求导致的时间浪费得以解决。

引入代理服务器的方案,相对于原始方案来说,主要的改进体现在以上两个方面。考虑到翻译服务的实现是依靠第三方翻译服务器,改进后的网页实时、在线翻译系统同样存在着第三方翻译服务器相对不稳定性和同第三方翻译服务器之间网络时延的问题。不过这对于客户端而言变得透明,代理服务器可以在不影响用户的情况下改变所使用的翻译服务器,这一定程度上缓解了传统方案中的客户端操作复杂度高的缺陷。

在改进方案中,影响 FGT 的主要因素同原始方案是一致的。而对于 PST 来说,如果重复请求同一个已经由代理服务器缓存的网页内嵌对象,则不再向目标站点进行请求,而是直接由代理服务器返回。因此在请求同一网页或者甚至是相同站点的类似网页时,PST 在重复请求中将会显著降低。

3 实验描述及测试结果

在现有的实验环境下,为了使实验更具有可控性,选取 Red hat Linux 操作系统作为平台。在 Linux 系统下,使用代理服务器软件 squid2.5 - STABLE11,安装在网关服务器上,提供网络上的高速缓存^[8]。同时选取的第三方翻译服务提供者连接速度较快的雅虎在线翻译。

各机器的作用和配置情况如下:

Gateway:作为 Client 访问外网的接口,对内网的所有客户机提供透明代理。当测试原始方案时,只起到普通的数据包转发工作,而测试改进方案时运行增加了内容协商机制的代理软件 squid。

Client:普通 PC。客户端使用浏览器 IE6.0,并且使用 Ethereal 0.10.14 监听网卡数据包,以获取时间相关数据。

测试的内容:在相同时间段,对于相同的一组 URL 列表(选取了国外网站 www.cnn.com 的 39 个英语新闻页面)中的页面进行原文浏览及翻译后浏览,同时选择通过代理服务或者不通过代理服务。分别得到相应的 FGT,PST,T。

(1)在原始方案下,不通过代理服务,直接使用第三方翻译器提供的服务。通过浏览器在第三方翻译服务的网页中粘贴要求进行翻译的目标网页的链接,然后从请求发出后开始直到浏览器呈现翻译后网页主要内容后设置浏览器的停止请求键。在监听客户端网络

数据的 Ethernet 软件所得的数据中,取得客户端从发出请求到获取已翻译网页的时间,以及从获取已翻译网页到浏览器停止发出请求的时间,前者即为 FGT,后者为 PST,而总时间为 T 。所得数据如图 2 所示。

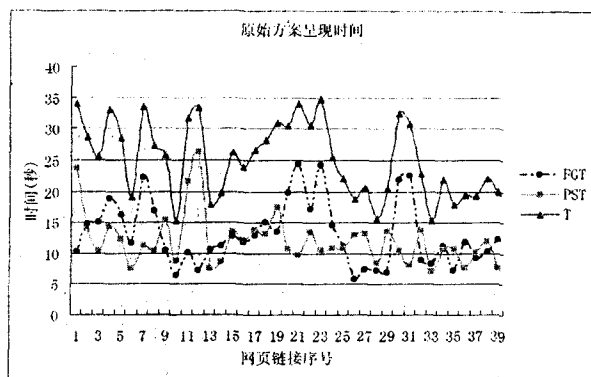


图 2 原始方案呈现时间

在传统网页在线翻译方法的呈现中,FGT 对于每个请求的网页链接的平均时间是 13.01 秒,PST 的平均时间是 12.17 秒。虽然国内对于国外网页的访问可能比较慢,但是 PST 的时间还是不尽如人意的。这还只是在不计人工操作时间的计算下得出的结论,计算人工操作时间将使效率更低。

(2)在改进方案中,对于 URL 列表中的每一个链接,通过代理服务器向第三方翻译服务器先后发出两次重复的翻译请求,在第一次请求完成后,接着再进行第二次请求。然后清空网关以及本地的缓存后,接着对 URL 列表中的下一个链接进行翻译请求。使用和原始方案相同的方式来获取数据,分别记录两次请求的实验数据,如图 3 和图 4 所示。

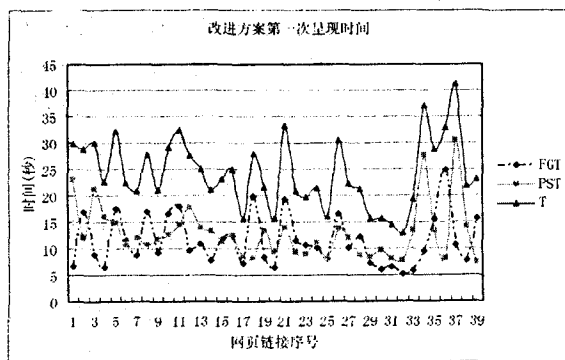


图 3 改进方案第一次呈现时间

由图 3 和图 4 的数据比较可以发现,第一次同重复请求的 FGT 相比几乎是差不多的,平均时间分别是 11.42 秒和 10.99 秒。而第一次请求的 PST 却几乎是重复请求的 2 倍,平均时间分别是 12.88 秒和 6.03 秒。这说明重复请求中代理服务的缓存功能使得重复请求网页对象的时间大大缩短。

(3)将改进方案的 FGT 同原始方案的 FGT 进行

比较(如图 5 所示)可以看出,由于代理服务并没有对网页的翻译效率做出实质贡献,除去 Internet 网上难免的时间不稳定性因素,两种方案取决于翻译效率的 FGT 相差很小,原始方案、改进方案第一次、第二次请求的平均 FGT 分别是 13.01 秒、11.42 秒和 10.99 秒。再将上述两方案的 PST 进行比较。由图 6 可见,对于同一网页甚至是同一站点具有类似网页对象的网页的重复请求,在代理服务的作用下,PST 将显著减小。

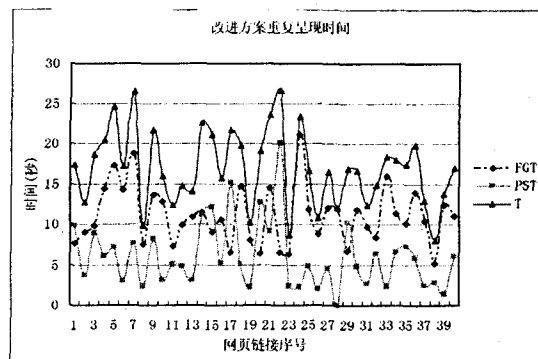


图 4 改进方案重复呈现时间

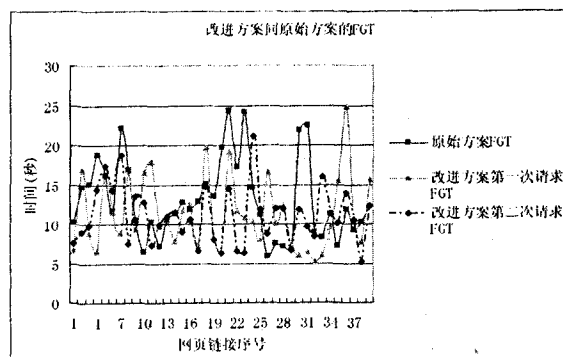


图 5 改进方案 FGT 同原始方案 FGT 比较

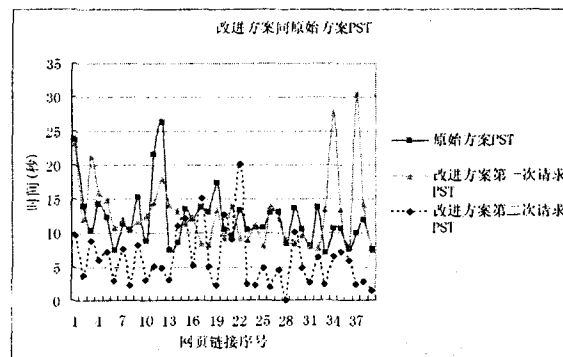


图 6 改进方案 PST 同原始方案 PST 比较

4 结束语

文中,首先对于当前最为常见的网页在线翻译方式——基于浏览器操作的在线翻译进行了说明和分析。在此基础上,提出了网络缓存(Web Cache)技术结

(下转第 16 页)

PIRP

```

Irp = IoBuildDeviceIoControlRequest( IOCTL_ INTERNAL_ USB_
SUBMIT_ URB, pdx -> StackDeviceObject, NULL, 0, NULL, 0,
TRUE, &event, &iostatus);
PIO_ STACK_ LOCATION stack = IoGetNextIrpStackLocation
(Irp); /* 获取 I/O 堆栈 */
stack -> Parameters. Others. Argument1 = (PVOID) urb; /* 把
URB 的地址填入 I/O 堆栈的 Parameters. Others. Argument1 域
*/
NTSTATUS status = IoCallDriver( pdx -> StackDeviceObject,
Irp); /* 发送请求到下一层驱动程序 */

```

由于控制命令的数据较少,可用 `_URB_ CONTROL_ VENDOR_ OR_ CLASS_ REQUEST` 类型的 URB 来完成,此 URB 可用宏 `UsbBuildVendorRequest` 构造,若在某些特殊场合需要自己构造一个控制传输的 URB(手动创建)时,其代码应具有如下格式:

```

|PURB urb=NULL;
urbSize=sizeof(struct _URB_ CONTROL_ TRANSFER);
urb=ExAllocatePool(NonPagedPool,urbSize);
RtlZeroMemory(urb,urbSize);
urb->UrbHeader.Length=sizeof(struct _URB_ CONTROL_
TRANSFER);
urb->UrbHeader.Function=URB_ FUNCTION_ CONTROL_
TRANSFER; //功能代码
//设置其他域
urb->UrbControlTransfer.PipeHandle=pipehandle;
.....
|

```

(上接第 12 页)

合 HTTP 协议所包含的内容协商机制,构建网页实时、在线翻译系统的具体方案。该方案对于传统网页在线翻译的不便和效率低下起到了改进的作用,内容协商机制的实现使得网页的翻译操作以及具体服务提供者对于客户端透明,使得使用浏览器浏览的客户端得以省去大量的不必要的操作时间。

通过一系列实验来比较传统网页在线翻译和基于内容协商和网络缓存网页在线实时翻译在网页呈现方面的性能。证实了改进方案的网络缓存技术对于网页内嵌对象,尤其是对于静态图像文件的缓存,使得改进方案相对于原始方案在重复请求相同网页或者同一站点类似网页的呈现效率上有了很大提高。实验数据证明改进方案重复请求的 PST 几乎是原始方案的 PST 的 50%。

参考文献:

[1] 王少春,陈家骏,王启祥,等. Internet 在线翻译浏览器技术

3 DDK 开发环境构造

要用 DDK 开发基于 WDM 的 USB 驱动程序,需要安装 VC++ 和 DDK,DDK 可从微软的网站上获得,需要注意的是在安装 DDK 前应该先完成 VC++ 的安装。之后可运行 `setenv.bat` 来设置 build 环境,也可手动添加环境变量来进行设置^[5]。一切就绪后,即可进行 WDM 驱动程序的开发。

4 结束语

介绍了基于 WDM 的驱动程序的基本结构,同时对开发过程中的疑难问题进行了深入剖析。在此基础上,介绍了 USB 的通信模型,分析了基于 WDM 的 USB 驱动开发的关键所在,给出了相应例程,并介绍了 DDK 开发环境的构建,最终结合实际系统完成了基于 DDK 的 USB 接口 WDM 驱动开发和调试。

参考文献:

- [1] 武安河. Windows 2000/XP WDM 设备驱动程序开发[M]. 第 2 版. 北京:电子工业出版社,2005.
- [2] Oney W. Programming the Windows Driver Model[M]. US: Microsoft Press,1999.
- [3] Microsoft Corporation. DDK Documentation[CP/DK]. 2003. <http://www.microsoft.com>.
- [4] USB - IF. Universal Serial Bus Specification Revision 2.0 [EB/OL]. 2002. <http://www.usb.org>.
- [5] 张弘. USB 接口设计[M]. 西安:西安电子科技大学出版社,2002.

探讨[J]. Application Research of Computers, 2001(1): 11 - 13.

- [2] Hao Mao. On Applied Machine Translation Softwares & Web Sites[J]. Chinese Science & Technology Translators Journal, 2004, 17(4): 24 - 25.
- [3] Wang Jia. A Survey of web Caching Schemes for the Internet [J]. Computer Communication Review, 1999, 29(5): 36 - 46.
- [4] Krishnamurthy B, Mogul J C, Kristol D M. Key Differences between HTTP/1.0 and HTTP/1.1[M/OL]. 1999. <http://www.research.att.com/bala/papers/h0vh1.html>.
- [5] Fielding R, Gettys J, Mogul J C, et al. Hypertext Transfer Protocol - HTTP/1.1 RFC2068[S/OL]. 1997 - 01. <http://rfc.net/rfc2068.html>.
- [6] Fielding R, Mogul J C, Frystyk H, et al. Hypertext Transfer Protocol - HTTP/1.1. RFC 2616[S/OL]. 1999. <http://rfc.net/rfc2616.html>.
- [7] Holtman K. Transparent Content Negotiation in HTTP - RFC 2295[S/OL]. 1998. <http://rfc.net/rfc2295.html>.
- [8] Wessels D. Squid: The Definitive Guide[M]. [s.l.]: O'Reilly, 2004.