

HTML5——下一代 Web 开发标准研究

刘华星 杨 庚

(南京邮电大学 计算机学院 江苏 南京 210003)

摘 要: HTML(HyperText Mark-up Language) 是目前网络上应用最为广泛的语言,也是构成网页文档的主要语言。旧有的 HTML 版本是 10 年之前制定的,在表达和应用上已经很难满足开发人员和用户的需求。为了适应这快速发展的 Web 技术以及不断提出的 Web 应用需求,HTML5 应运而生。首先简要描述了 HTML5 的发展历程,然后重点介绍 HTML5 的特点和应用,并与现有的 Web 技术如 RIA(Rich Internet Applications)、cookies 等在网络通信、数据存储、地理定位、音视频播放、网页绘图方面进行比较,分析 HTML5 在这些领域的优势,从而得出结论。HTML5 较之前的版本加入的新功能,能够使开发和使用网页更加便捷,并与浏览器很好地结合,降低了浏览器对资源的占有率以及对插件的依赖。最后探讨了 HTML5 中存在的一些问题和今后主要发展的方向。

关键词: HTML5; RIA; flash; Web Socket; Offline Storage

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2011)08-0054-05

HTML5-Next Generation Web Development Standard Research

LIU Hua-xing, YANG Geng

(College of Computer Science, Nanjing University of Post and Telecommunications, Nanjing 210003, China)

Abstract: HTML (HyperText Mark-up Language), which is the most widely used Web language in the Internet, is the main language that forms Web documents. The old version proposed by 10 years ago, has been difficult to meet the needs of developers and users in the expression and application. HTML5 came into being, in order to adapt to the rapid development of Web technology and the demands of Web applications. It introduces the development process of HTML5 at first, then describes the characteristics and application of HTML5, and compares it in the factors of network communications, data storage, geographic location, and canvas with the existing Web technologies, such as RIA (Rich Internet Applications), cookies and so on, in addition to be analysis of the superiority of HTML5 in these areas. So it is the conclusion that compared with the previous versions, HTML5 adds many new features, which make it easier to develop and use the Web, better to integrate with the browser, less to share resources in the browser and dependence on the plug-in. Finally, discuss problems existing in HTML5 and the direction of development in the future.

Key words: HTML5; RIA; flash; Web Socket; Offline Storage

0 引言

在 HTML4 推出后的十年里,各组织机构都开展了一些下一代 Web 应用协议的制定工作。这期间,最受关注的是 WHATWG (Web Hypertext Application Technology Working Group) 工作组开发的 Web Application 1.0 草案,该草案于 2007 年被 W3C 接纳,成为 HTML5 的前身。2008 年 1 月 22 日,第一份正式 HTML5 草案公布,并于 2010 年 9 月正式向公众推荐,并有望在 2012 年发布。

目前在 PC 上,Safari、Chrome、火狐、遨游和 Opera 浏览器都在不同程度上支持 HTML5,其中 Chrome 支持度是最好的。在手机平台上,苹果 iPhone、谷歌 Android 以及 Windows Mobile 上的 Zetakey 也支持 HTML5 的许多功能。而在微软最新推出的 IE9 中也加入了对 HTML5 的支持,这将更进一步推动 HTML5 的推广与应用。在网页应用上,国外的 YouTube 以及国内的 YouKu 等主要音视频网站都已经开始支持 HTML5 形式的音视频媒体。此外在许多网页上也实现了绘图、动画等一些在 RIA 中才会出现的应用,其实现的效果并不比用 RIA 实现来的差。而 HTML5 之所以能如此迅速的发展,得益于它的新标签和新特性。

相较于 HTML4,HTML5 在标签元素上增加了新的特殊内容元素,比如 article、footer、header、nav、section;新的表单控件,比如 calendar、date、time、email、

收稿日期:2011-01-04;修回日期:2011-04-16

基金项目:国家自然科学基金资助项目(60873231);江苏省自然科学基金(BK2009426)

作者简介:刘华星(1987-),男,硕士研究生,研究方向为多媒体平台应用技术;杨庚,博士后,教授,博士生导师,研究方向为计算机通信与网络、网络安全、分布与并行计算等。

url、search; 新的页面元素, 比如 header、section、footer、figure。同时还废除了 basefont、big、center、font、s、strike、tt、u、frameset、frame、noframes、plet、bg sound、blink、marquee 等元素^[1]。在功能特性上 HTML5 中加入了一些新特性, 如语义化的标签、音频和视频的直接支持、矢量绘图、支持网页内容编辑、Web Socket、地理定位、多线程处理以及离线存储等。

文中综述了 HTML5 新增的标签和特性在网络通信、数据存储、地理定位、音视频播放、网页绘图等方面的应用, 通过与现有的 Web 技术的比较, 探讨了其特性在研究应用上的优劣, 展望了未来的研究方向。

1 HTML5 的新特性及应用

1.1 Web Socket

Web Socket 是 HTML5 规格中的一个非常重要的新特性, 它的存在可以允许服务器与客户端、浏览器之间实现双向连接, 这个连接是实时的, 可以实现数据的及时推送, 并且该连接持续开放直到明确关闭它为止^[2]。这就使得人们可以通过网页实现很多以前无法实现的应用。例如在线聊天应用, 如果用户想要发送一句话, 在按下“发送”按钮的同时, 浏览器会递送给目标服务器一个请求, 然后服务器端会将这句话显示在网页上。而用户如果想要看到这句话, 他需要刷新自己的浏览器以获得最新页面才可以。而通过 Web Socket 技术构建的 HTML5 应用, 用户不再需要频繁的刷新页面以获取新数据, 新数据会自动由服务器端推送至用户的电脑屏幕上^[3]。这样就可以用网页做到许多即时聊天工具如 MSN、QQ 等的效果。相较于这些聊天工具来说, 用户无需在 PC 端装上客户端, 只需要有支持 HTML5 的浏览器即可。

文献[4]介绍了在 Web Socket 出现以前, 要实现网页通信的效果主要是通过轮询和其它服务器端推送技术实现的。最著名的是 Comnet, 它使用各种不同的 hack 技术, 推迟了 HTTP 响应的结束。基于 Comnet 的推送通常是使用 JavaScript 结合长轮询(Long Polling)或流连接策略实现的。这两种策略都要在数据头部包含额外的信息, 降低了网络利用率; 还要同时构建上行和下行通道连接, 在维护和协调这两个连接上需要大量的系统开销, 并增加了复杂性。

Peter Lubbers 和 Frank Greco 在文献[5]中提出, 与长轮询和流连接相比, HTML5 的 Web Socket 定义了一个全双工的通信通道, 通过一个单一的套接字在 Web 上进行操作, 从而减少了不必要的网络流量与延迟。它使得数据字节数最大可减少到 2 字节, 延迟从 150 毫秒减少到 50 毫秒。Web Socket 在连接建立使用上也比长轮询和流连接来的便捷。首先, 在 Web Sock-

et 建立的时候是通过将客户端与服务器之间 HTTP 协议的初始握手阶段升级到 Web Socket 协议来建立的, 而其底层仍是使用 TCP/IP 连接。其次, 在连接建立成功后, 它是通过使用标准 HTTP 端口(80 和 443)进行通信。因此, HTML5 Web Socket 不需要安装新的硬件, 或者要求公司网络开放新的端口。在浏览器与 Web Socket 服务器之间不需要任何中间服务器(代理或反向代理服务器, 防火墙, 负载均衡路由器等等), 只要服务器和客户端双方都理解 Web Socket 协议, Web Socket 连接就可以顺利地建立。

1.2 地理定位

现今网络中定位用户的位置的技术主要是通过 IP 地址来探测的。HTML5 的地理定位是一个精确定位用户的替代方法, 其通过加入 geolocation 的 API 来实现, 使得 Web 第一次能真正在自己的领地里实现地理定位。该功能通过 getCurrentPosition 和 watchPosition 这两种方法实现。其中 getCurrentPosition 是用来获取用户当前位置, 而 watchPosition 则保持用户的位置, 并且按照常规的时间间隔持续查看用户的位置是否发生变化。如果用户位置发生变化则告知 getCurrentPosition 做出改动。getCurrentPosition 函数调用成功时会进入回调 success 函数, success 函数有一个参数是 position 对象, 这个 position 对象有一个 coords 对象, coords 对象包含了很多地理位置信息, 如 latitude(纬度)和 longitude(经度), 这样就可以知道你的具体位置了^[2, 4, 6]。这个功能在一些手机如 Iphone 上已经有广泛的应用了。

文献[2]指出 HTML5 在获取具体位置上是通过 GPS 或网络信息来获取。GPS 虽然是获取信息最显著的方法之一, 但是在室内或是周围遮挡较多的时候, GPS 就难以使用, 这时通过网络方式来获取信息就是一个很好的方案。它是根据网络信号如 IP 地址、RFID、WiFi 及蓝牙 MAC 地址推断出位置信息的。具体来说就是运行 HTML5 的浏览器会去获取有效范围内检测到的无线热点对应地址, 然后将这些无线热点信息及电脑本身的 IP 地址、网卡 Mac 地址提交到服务供应商。在服务供应商的数据库中保存着固定、持久的无线热点信息。它会根据浏览器提交的请求信息, 对比搜索对应的无线热点信息, 然后通过 WiFi 三角测量等方法将最近的那个无线热点的坐标位置返回。与此同时服务提供商也把新的 MAC 地址加入那个数据库, 形同把用户的手机或电脑转化成一个 Wi-Fi 感应器。

该功能最显著的应用是在手机上, 通过支持 HTML5 的浏览器获取定位信息, 就能定位到两三米的距离, 这对手机用户来说拥有很大的价值。通过 HTML5

定位更有价值的地方是这种定位方式除了可以定位自己的位置,还可以获取到他人的定位信息,当然这种信息的获取是要在他人对你开放信息的前提下。此外,Web 服务提供商还可以利用该功能为用户提供用户间的互动类服务。

1.3 数据存储与离线 Web 应用

传统的客户端存储方式有 cookie、Firefox 下的 globalStorage、Flash 插件自己的存储方式,但是这几种方式都有其局限性(安全性和兼容性)。以 cookie 存储为例,大多数浏览器对 cookie 的限制最多不能超过 4096 个字节(4kB),且 cookie 的数量总共不得超过 300 个,单个域或本地的 cookie 不得超过 20 个,这样的限制已经完全无法满足如今的 WEB2.0 站点的需求。

因此 HTML5 提出了自己的数据存储方式,文献[7]中说明了其两种方法:

1) Web SQL 数据库(Web SQL databases),它将 Web 数据存储为数据库形式,开发者可以基于 SQL 语法来对这些数据进行查询、插入等操作。

2) 客户端存储方式(Web Storage),它在实现上又有共两种方式:

(1) 局部存储(Local Storage)。

(2) 会话期存储(Session Storage)。

Local Storage 可以永久保存数据,即对支持 HTML5 本地存储的浏览器会在用户的本地分配空间用于永久性存放指定的数据,在用户关闭浏览器后,下次打开浏览器还能取得存储的数据,除非用户主动删除这些数据。此外这些数据是在同一个域名下共享的。它的存储对象是以键/值对形式存放的。

Session Storage 只在当前的会话中可用,即在用户关闭窗口后,数据将清除,此外这种存储方式在同域中无法共享。它的存储对象也是以键/值对形式存放的。

HTML5 的 Web Storage 两种存储方式所存储的数据是不能跨浏览器读取的,即用一种浏览器打开站点保存的数据,用其他浏览器是取不到的。而且这两种方式存储的数据在用户浏览期间不能够像 cookie 一样传送到服务器,如果需要把数据发送服务器,必须要使用 XMLHttpRequest 对象做异步传输^[4 8]。

与 cookie 相比,HTML5 的 Web Storage 存在以下几点优势:

1. 存储空间更大:它的存储空间可以达到 5MB。这在浏览器上的实现,足以取代 4kB 的存储容量的 cookies。

2. 存储内容不会发送到服务器:当设置了 cookie 后,cookie 的内容会随着请求一并发送到服务器,这对

于本地存储的数据是一种带宽浪费。而 Web Storage 中的数据则仅仅是存在本地,不会与服务器发生任何交互。

3. 更多丰富易用的接口:Web Storage 提供了一套更为丰富的接口,使得数据操作更为简便。

4. 独立的存储空间:每个域(包括子域)有独立的存储空间,各个存储空间是完全独立的,因此不会造成数据混乱。

在实际应用中,HTML5 这个特性使得当程序需要处理大量的数据时,避免数据频繁地在客户端和服务器的往来。这对移动设备来说,可以极大地减少流量的消耗。而这个特性最大的应用却是在于用户在没有网络连接的时候,例如火车驶入隧道的情况下,可以通过本地缓存正常使用 Web 应用。当连接到网络后,再同步到 Web 上,并自动在这些本地缓存的副本改变时更新,这就是 Web 的离线应用^[9]。

1.4 多线程支持

由于 JavaScript 不是一种支持并发编程的语言,所以长期以来 Web 应用程序被局限在一个单线程世界中。同时 Web 页面已经发展成为 Web 应用程序,使用 JavaScript 完成的任务的复杂程度已经大大增加,这些复杂的 JavaScript 存在 UI 应用程序死锁的风险。HTML5 利用 Web Workers 将多线程引入 Web 应用程序,使这些复杂的应用成为可能。

Matthew David 在文献[10]中介绍了 Web Workers 及其应用。Web Workers 提供了一种标准的方式让浏览器能够在后台运行 JavaScript。有了它,就可以通过创建一个 Worker 对象实现多线程了。页面动态加载 JavaScript 文件之后,它就可以在后台处理诸如复杂的数学运算、发送请求或者操作本地数据库这样的代码,而不影响用户界面和响应速度,与此同时页面还可以响应用户的滚屏、单击或者输入操作。这就使得那些需要长时间运行的脚本不会被用户点击或交互而运行的脚本中断,此外还可以不必为考虑页面响应的问题,而去终止对于那些需要长时间执行的任务。在后台处理结束相应的任务后,前台只要使用 callback 函数接收处理返回的数据即可。此外还要注意后台运行的 Worker 不能操作 DOM,如果希望后台程序处理的结果能够改变 DOM,只能通过返回消息给创建者的回调函数进行处理。后台的线程与线程之间以及线程与界面之间只能通过消息事件通信。

Ian Hickson 在文献[11]中指出另有一种类型的 Web Worker 是 Shared Worker,当前只在 Chrome 和 Webkit 中支持。一个 Shared Worker 实际上很像是一个普通的 Web Worker,只不过多个文档可以访问该 Worker 相同的单个实例。这意味着,如果存在几个弹

出窗口或者几个 `iframe` ,所有这些文档可以访问这个单个的 `Shared Worker` ,并且这个单个的 `Shared Worker` 将为所有那些文档提供服务。例如在应用中会打开同一个应用的多个网页窗口 ,而此时该应用的客户端需要升级 ,就可以使用 `Shared Worker` ,只需对客户端进行一次升级操作而不是多次。

1.5 播放器

HTML5 提供了 `video` 和 `audio` 标签。它允许开发者直接将视频和音频嵌入网页 ,不需要任何第三方插件(比如 Adobe 公司的 `Flash`) 就能播放。目前 ,支持 HTML5 的浏览器在支持音频视频格式上并没有统一的标准 ,浏览器厂商是自行选择支持的格式。

- 其中 `video` 元素主要支持三种视频格式:
- 1) `Ogg` 格式 ,即带有 `Theora` 视频编码和 `Vorbis` 音频编码的 `Ogg` 文件;
 - 2) `MPEG4` 格式 ,即带有 `H.264` 视频编码和 `AAC` 音频编码的 `MPEG 4` 文件;
 - 3) `WebM` 格式 ,即带有 `VP8` 视频编码和 `Vorbis` 音频编码的 `WebM` 文件。

- `audio` 元素支持三种音频格式:
- 1) `Ogg Vorbis` 格式;
 - 2) `MP3` 格式;
 - 3) `Wav` 格式。

表 1 和表 2 是各浏览器对 HTML5 音频、视频上支持的情况。

表 1 各浏览器对 HTML5 视频格式的支持			
	MPGE4	Ogg	WebM
IE 9	支持	不支持	支持
Chrome 6	支持	支持	支持
Opera 10.6	不支持	支持	支持
Safari 5	支持	不支持	不支持
Firefox 4	不支持	支持	支持

表 2 各浏览器对 HTML5 音频格式的支持			
	MP3	Ogg Vorbis	Wav
IE 9	支持	不支持	支持
Chrome 6	支持	支持	不支持
Opera 10.6	不支持	支持	支持
Safari 5	支持	不支持	支持
Firefox 4	不支持	支持	支持

音频视频格式之所以无法统一 ,主要是由于各浏览器厂商之间竞争引起的。以视频编码方式为例。因为苹果公司和微软公司都是 `MPEG-LA` “专利池”的所有者 ,所以就支持推广以 `H.264` 编码的 `MPGE4` ,而其他厂商则使用 `Theora` 这种开源的、无专利的编码方式或 `Google` 推出的开源的 `V8` 模式^[12]。

这些音频视频嵌入网页的使用直接冲击了以 `Flash`、`Silverlight`、`JavaFX` 为代表的富互联网应用 (`RIA`) 。

首先 ,对于 `RIA` 就要在页面中包含插件 ,此时页面就需要保留某一个绘制区域给插件。对浏览器来说该区域始终是一个黑盒 ,浏览器不会处理或解释在其中的任何事情。当页面布局和插件的绘图区域重叠或是页面有动态的布局变化的时候 ,都会引发问题^[2]。而 HTML5 标签通过 `CSS` 样式处理和 `JavaScript` 的操作 ,一切都是利用浏览器内建功能完成。

其次 ,这些插件容易成为浏览器不稳定的重要原因 ,同时随着浏览器中的插件的增多 ,要跟踪每个浏览器插件中可能存在的安全漏洞越来越复杂 ,在安全上带来问题。而 HTML5 把这些功能内置到而不是使用插件可以大大降低安全风险 ,避免与插件开发有关的多个环节出现问题 ,更可以防止有人故意利用插件中的 `API` 安装恶意代码^[13]。而且 HTML5 是由多组织共同制定的 ,相较于插件如 `Flash` 只由 Adobe 一家公司控制的封闭式系统来说 ,在漏洞、安全处理上要严密的多。

再次 ,HTML5 要比 `RIA` 技术占用更少的系统资源 ,特别是在移动设备使用上 ,可以通过硬件解码支持视频播放 ,提高了性能 ,节省了资源使用以及电池的电力消耗^[14]。移动设备相同的电力上通过 `Flash` 和 `HTML5` 播放视频的时间比可达到 1 : 2。

最后 ,基于 HTML5 的音频、视频或 Web 应用程序 ,相较于使用 `RIA` 开发的具有更好的移植性 ,只要有支持 HTML5 的浏览器存在就可以播放运行 ,无论是在 `Windows`、`Unix`、`Mac OS` 以及手机平台上。

当然 `Flash` 对于 HTML5 也存在优势 ,在音频视频上它的视频旋转、音频视频会议和录音、视频动态覆盖对象等都是 HTML5 目前无法实现的^[15]。此外 ,在其他功能应用上还有其独特之处 ,如多点触摸、人脸识别、`Socket` 通讯同步、`AR` 增强现实等。所以 ,未来 HTML5 可能会作为一个承载的平台 ,和 `RIA` 相结合共同运用。

1.6 画布

HTML5 提供了 `Canvas` 标记元素来实现画布功能。该元素可以使用浏览器脚本语言(通常是 `JavaScript`) 进行图形绘制 ,例如可以用它来画矢量图 ,合成栅格图 ,或者绘制复杂的动画以及文本文字 ,这些图形将直接渲染在浏览器上。相较于传统的在服务器端先画好图片 ,再把图片发到浏览器中 ,用 `Flash` 或其他第三方插件显示的方式 ,它与浏览器渲染引擎紧密结合 ,节约了资源 ,并极大地简化了图形和网页中其他元素的交互过程。可是 HTML `Canvas` 只提供了矩形和路径的绘

制,不直接提供点、线、多边形的绘制,所有其它复杂的图形都是通过一些函数来组成的^[16]。

画布绘制在实现上可以有两种方式:其一,通过 Canvas 元素的“上下文(rendering context)”,类似于绘图板上的一页,在其中可以绘制任意 2D 图形或动画;其二,通过 WebGL 技术来实现 3D 图形动画的绘制。

HTML5 的画布功能相较于使用 XML 文档来描述绘图的 SVG 技术,各有其优势和适合应用的地方,Pe-dro Rogério 在文献[17]做了一定的比较:

1) 画布是基于位图的图像,它不能改变图形大小,只能缩放显示,要改变图像就要对每个像素进行操作。SVG 运用简单高效的矢量图形指令代替对每个像素信息的描述,图形在被按任意比例缩放时都能保证图形被缩放局部的清晰,没有光栅图常见的锯齿现象。

2) SVG 存在 DOM 树,它表示绘制到屏幕上的所有对象的当前状态,这使得它成为一种保留模式的 API。由于这棵树可用,它就拥有很好的交互性,可以绑定外部点击或接触事件到树中的特定节点上,并且监听这些事件,待事件发生就可以用脚本来响应。而画布是一个低层级的 API,它不存在 DOM 树,在对鼠标点击交互处理上效果不好,却适合对键盘交互的操作。

3) 对于绘制的文本,SVG 绘制的文本可以像普通文本一样操作,而在画布上绘制的文本却不能从图像中抓取分离出来,也不能被搜索引擎获取。

4) 画布适合对拥有大量对象的图像进行绘制,而 SVG 适合对有大区域范围的图像,如地图,进行绘制。

5) 画布作为一个元素标记可以和 HTML 很好地结合,而 SVG 虽然可以很好地应用在 XHTML 中,却很难直接应用于 HTML。

2 安全问题与不足

HTML5 加入了这些新特性的同时也引入了一些安全上的问题。在 HTML5 问世前,攻击者可能不得不窃取电脑上的 cookie,破解之后获得用户的密码,进而读取或植入数据。而现在,由于存储在本地的数据未加密而且永远不会过期,攻击者只要能进入用户的浏览器就能在使用者不知情的情况下,读取来自浏览器的数据,或者把其他数据植入到浏览器中。尤其是在使用公共电脑的时候,这些存储在本地的数据极易造成个人隐私的泄露。借助地理位置功能,攻击者就能在用户不知情的情况下确定用户的位置。Web Socket 为浏览器提供了网络通信堆栈,攻击者利用这项特性,可以轻易地进行后门通信^[18]。此外,HTML5 跨平台支持也为网络攻击者提供了更多机会,现在攻击者不

仅可以窃取用户 PC 上的资料,通过手机上支持的 HTML5 浏览器,攻击者还能对用户手机进行攻击。

HTML5 作为一个尚不成熟的协议,除了引入安全隐患外,还存在着以下几点不足:

- (1) 不能进行网络摄像头的交互;
- (2) 不能通过网络摄像头进行录制;
- (3) 不能通过麦克风录音;
- (4) 不能做任何种类的在线会议;
- (5) 视频不能在 3D 水平上使用;
- (6) 不能添加动态对象到视频上,例如字母、标题、导航,特别是时间轴字幕;
- (7) 不能创建桌面应用程序;
- (8) 还未支持 Peer-2-Peer;
- (9) 不支持全屏模式。

毕竟 HTML5 仍处于完善之中,相信到 HTML5 正式发布的时候这些缺陷与不足之处都将被解决,呈现出来的会是一个功能完善、安全可靠的新标准。

3 结束语

根据上述分析,HTML5 引入了许多新的特性和功能,丰富了 Web 应用,当然也存在着不足与缺陷。而随着其标准的进一步完善,各浏览器对其的支持度也将进一步提高,各网站也会逐渐从当前模式向 HTML5 升级。将来 HTML5 除了在解决安全机制问题和功能缺陷上还将在以下几点被更加深入的研究和应用:

1) 借助于 HTML5 功能特性,以其为基础开发 WebOS(Web 操作系统),实现操作系统的更新换代。

2) 在手机等移动设备领域,支持 HTML5 的浏览器将迅速增长,改善移动网络的应用。

3) 随着三网融合的推进,HTML5 有可能成为未来电视平台系统的一部分,使未来的电视也能够观看网络视频^[19]。

4) HTML5 将成为未来 Web 开发的平台,而 RIA+HTML5 的应用可能成为未来 Web 应用的主导,带来更加丰富的网络应用。

5) 在实现 Web3.0 上,HTML5 也能够做到很好的支持。

参考文献:

- [1] W3School. HTML5 教程[EB/OL]. [2010-10-28]. <http://www.w3school.com.cn/html5/>.
- [2] Lawson B, Sharp R. Introducing HTML5 [M]. United States of America: Pearson Education 2010.
- [3] White Paper. Towards Real-time Interaction on the Web [EB/OL]. 2010-03-12 [2010-11-05]. http://pulsarlabs.org/files/2010-03-12_PulsarLabs_White_Paper.pdf.

(下转第 62 页)

统资源定位性能的一个重要指标,仿真实验结果表明兴趣子网中的平均资源定位路径长度随着网络规模的增大也是服从对数关系的。

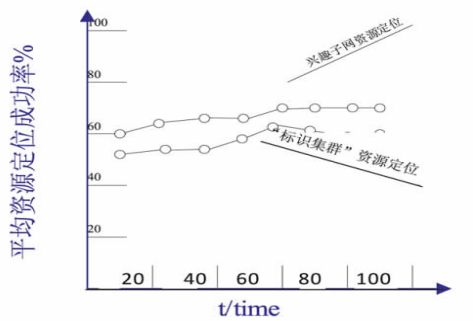


图 4 资源定位成功率的比较

5 结束语

从兴趣子网的角度出发,构建了资源子网,详细分析了该算法,在 Freenet 中对使用资源子网进行初始路由,资源定位做了深入的研究,通过仿真实验对原有算法和改进算法进行了比较,得出基于兴趣子网的算法提高了 Freenet 网络资源定位的性能。

参考文献:

[1] Luo Xucheng, Qin Zhiguang. Efficient multi-source location in unstrucee P2P systems[C]//International Conference on Parallel Processing Workshops. [s. l.]: [s. n.] 2007.

[2] Zhao B Y, Huang Ling, Stfib Ling J, et al. Tapestry: aresilient global-scale overlay for service deployment [J]. IEEE Journal on Selected Areas in Communications, 2004, 22(1): 41-53.

[3] Tateno M, Shimizu S, Arakawa Y, et al. Construction of overlay network considering user preference in peer-to-peer systems [J]. Technical Report of IEICE, 2006, 105(2): 143-146.

[4] 赵宏, 谢伟志, 张晨曦. 基于蚁群算法的非结构化 P2P 搜索研究[J]. 计算机技术与发展, 2009, 19(2): 31-34.

[5] 高飙, 徐家品. 基于兴趣子网的 Gnutella 网络资源定位的研究[J]. 计算机工程与设计, 2009, 30(17): 3992-3995.

[6] 蔡诗维, 钟诚. 基于兴趣社群的 P2P 网络节点自治管理方法[J]. 计算机技术与发展, 2008, 18(3): 161-164.

[7] 薛广涛, 贺小箭, 贾兆庆. 使用兴趣子网划分算法对 Gnutella 中资源定位机制的改进[J]. 上海交通大学学报, 2004, 38(12): 2108-2110.

[8] 陈贵海, 李振华. 对等网络: 结构、应用与设计[M]. 北京: 清华大学出版社, 2007: 63-81.

[9] 夏启志, 谢高岗. 无结构 P2P 网络搜索方法及其改进[J]. 计算机应用研究, 2005, 22(9): 256-260.

[10] 张国治, 党小超, 魏伟一. 基于信任域的 P2P 访问控制模型研究[J]. 计算机技术与发展, 2006, 16(8): 228-230.

[11] 李振华, 陈贵海, 邱彤庆. 分点: 无结构对等网络的拓扑关键点[J]. 软件学报, 2008(9): 2376-2388.

[12] 谷科, 危韧勇. 对基于 CAN 的 P2P 网络模型的改进研究[J]. 小型微型计算机系统, 2007(3): 2-5.

(上接第 58 页)

[4] Lubbers P, Albers B, Salim F. Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development [M]. United States of America: Apress, 2010.

[5] Lubbers P, Greco F. HTML5 Web Sockets: A quantum leap in scalability for the web [J]. Jay View, 2010, 21(1): 3-7.

[6] Hickson I. HTML5 标准文档[EB/OL]. 2010-10-19 [2010-11-13]. <http://www.w3.org/TR/html5/>.

[7] 王明超, 陈榕. 基于脚本的客户端软件结构化存储技术研究[J]. 计算机技术与发展, 2010, 20(2): 25-28.

[8] Pilgrim M. HTML5: Up and Running[M]. United States of America: O'REILLY, 2010.

[9] Leblon R. Building advanced, offline web applications with HTML5 [EB/OL]. [2010-12-06]. http://lib.ugent.be/fulltxt/RUG01/001/418/513/RUG01-001418513_2010_0001_AC.pdf.

[10] David M. HTML5: designing rich internet applications [M]. United States of America: Focal Press, 2010.

[11] Hickson I. Web Workers [EB/OL]. 2010-11-07 [2010-11-13]. <http://www.whatwg.org/specs/web-workers/current-work/>.

[12] Daoust F, Hoschka P, Patrikakis C Z. Towards Video on the

Web with HTML5 [EB/OL]. [2010-11-25]. <http://nem-summit.eu/wp-content/plugins/alcyonis-event-agenda/files/video-html5.pdf>.

[13] Pilgrim M. Dive Into HTML5 [DB/OL]. [2010-11-10]. <http://diveintohtml5.org>.

[14] 唐灿. 下一代 Web 界面前端技术综述[J]. 重庆工商大学学报: 自然科学版, 2009, 26(4): 350-355.

[15] Vannavada S, Haven B. Apple Instigates the Flash vs. Html5 Slam Dance [EB/OL]. [2010-11-16]. <http://www.icrossing.com/articles/Apple-Instigates-the-Flash-vs-HTML5-Slam-Dance.pdf>.

[16] 吴磊, 张福庆. 基于 HTML canvas 的 WebGIS 客户端技术研究[J]. 地理信息世界, 2009(3): 78-82.

[17] Rogério P. SVG VS Canvas [EB/OL]. 2010-09-01 [2010-12-11]. <http://www.pinceladasdweb.com.br/blog/2010/09/01/svg-vs-canvas/>.

[18] 沈建苗. 警惕 HTML5 引发新的安全问题[N]. 计算机世界, 2010-09-06(S07 版).

[19] 吴权, 赖斌. 浅谈新网络技术标准带来的改变[J]. 电脑知识与技术, 2010, 6(15): 3937-3938.