

面向瘦客户端的 WebOS 研究

任 怡 管剑波 尹虎哲 吴庆波 戴华东

(国防科学技术大学计算机学院软件研究所 长沙 410073)

摘 要 基于浏览器的 WebOS 是操作系统技术向网络化发展的一个典型代表。WebOS 整合 Internet 上服务和资源,供在线用户“随时随地”使用,这种特点使得它将成为未来瘦客户端 Web 应用的一种重要使用模式。首先阐述了 WebOS 的由来、概念和组成,接着分析了 WebOS 的发展现状和趋势,然后论述了制约其发展的主要技术因素。在此基础上,选择较有代表性的开源 EyeOS 进行了深入研究,分析了其原理和组成,并对其服务器端功能进行了扩充。

关键词 云计算,瘦客户端,WebOS,Web 服务

中图法分类号 TP316 文献标识码 A

Research on Thin-client Oriented Web Operating Systems

REN Yi GUAN Jian-bo YIN Hu-zhe WU Qing-bo DAI Hua-dong

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

Abstract With the development of distributed computing technologies, browser-based Web operating system (WebOS) came into being. It provides mechanisms for integrating services and resources on the Internet. And it makes services and resources available for online users regardless of where they are and when it is. WebOS represents a trend towards future use of thin client based Web applications. It has become one of the most important patterns of utilization in the coming cloud computing era. We first introduced the concepts and structure of WebOS. Then we analyzed related work and the trends of WebOS. And we also discussed the technical factors which cumber the development of WebOS. Finally, we made comprehensive research on EyeOS, which is a typical open source WebOS. We analyzed EyeOS's structure and APIs. Then we extended its functions by implement a SOAP-based Travel service.

Keywords Cloud computing, Thin client, WebOS, Web service

1 引言

2008 年以来,云计算技术得到了工业界和学术界的广泛关注。云计算作为一种资源交付和使用模式,支持通过互联网便捷地访问可定制的、快速部署的 IT 资源,并只需要较少的管理工作或较少的与服务供应商的交互。这里,提供资源的网络被称为“云”。“云”中的资源在使用者看来是可以无限扩展的,并且可以随时获取。目前,云计算平台自下而上分为 IaaS(基础架构即服务)、PaaS(平台即服务)和 SaaS(软件即服务)3 个层面,分别为用户提供原始的 IT 硬件资源和基本的操作系统环境、分布式和网络应用的开发和运行环境及基于基础设施的应用级服务。

随着移动终端、手持设备、个人电脑等瘦客户端的不断普及,AJAX、Web 2.0 等新技术的发展及在云计算随时随地获取资源的这一需求的驱动下,出现了一种新的网络资源使用模式——WebOS (Web-based Operating System)。尽管 WebOS 这一概念在 20 世纪 90 年代就已被 Berkeley 大学

WebOS 工作组提出^[1],但直到近几年才得到快速发展,其形态也发生了改变。WebOS 作为对云计算平台 SaaS 层面网络资源和服务整合的一种模式,体现了未来网络使用方式发展的重要趋势。

2 WebOS 的概念、组成和特点

2.1 WebOS 概念

GC FOX 等人于 1995 年提出了 WebWindows 的概念,他们认为 WebWindows 将作为一个开放的、公用的和模块化的 Internet 分布式操作系统^[2]。随后, Berkeley 大学的 WebOS 工作组首次提出了 WebOS 概念,并提供了一种全新的 Internet 服务范例,目标是为地理上分布的、高效的、可逐步扩充的、可动态重构的 Internet 广域分布式应用提供操作系统服务,进而使用户无缝地、透明地访问 Internet 上任意站点的任何资源^[1]。

目前,WebOS 通常由运行在浏览器中的程序集合及其管理和开发环境组成,通过客户端浏览器将分布在网络中的服

到稿日期:2009-07-02 返修日期:2009-10-12 本文工作受国家自然科学基金项目(60603063,90718040 和 60633050)和国家 863 项目(2006AA01A104,2007AA01Z177,2008AA01Z138 和 2009AA01Z101)资助。

任 怡(1977-),女,博士,助理研究员,主要研究方向为系统软件和中间件,E-mail:renxiaoyi@21cn.com, renyi@nudt.edu.cn;管剑波(1977-),男,助理研究员,主要研究方向为网络通信和系统软件;尹虎哲(1986-),男,本科,主要研究方向为系统软件和中间件;吴庆波(1969-),男,研究员,主要研究方向为系统软件;戴华东(1975-),男,副研究员,主要研究方向为系统软件。

务器端的各种应用程序和服务虚拟为本地操作系统服务,在浏览器中展现窗体或 Shell 风格的操作系统用户界面。以窗体界面为例,支持将文件管理、在线办公、日历、计算器、编辑器、聊天工具、播放器等功能模块以桌面图标方式激活,提供任务栏和开始菜单,从而模拟、代替或很大程度上补充桌面操作系统环境^[3]。只要用户的计算设备能够访问 Internet,并安装了浏览器,即可通过账号登录使用 WebOS,获得与本地操作系统类似的用户体验。从所处软件层次来看,WebOS 并不是传统意义上的操作系统,其实质是分布式应用程序的集成、管理和开发环境;从广义范畴来看,WebOS 为未来的网络使用模式提供了操作环境,支持包括用户管理、网络存储、文件管理、安全管理等在内的各种基本支撑库和服务,因此亦被称作 Web 操作系统。

2.2 WebOS 的拓扑组成

WebOS 采用 B/S 结构,主要由客户端浏览器显示的虚拟 OS 界面、客户端与服务器端的连接和优化(采用 AJAX、Applet、DHTML 等技术实现)及服务器端支持 3 部分组成。

客户端一方面支持将部分功能模块下载到本地进行,从而提高用户响应速度;另外,可接受用户输入,通过透明的传输协议发送给服务器,从而向服务器请求服务。服务器端包括一组可用的基于网页的桌面应用程序集合,存储、同步、安全等公共服务及应用/服务集成和开发环境等支撑机制。服务器端的硬件部分由多台机器和网络组成。WebOS 允许用户定制自己的客户端界面,保留所需的功能,这就需要服务器方提供的服务是模块化可配置的。WebOS 需要客户端和服务端共同提供技术支持。

3 WebOS 的现状和发展

WebOS 利用 Internet 中的存储和计算资源处理相应的运算,使得用户的数据运算不再受限于本地计算机的计算功能和存储能力。WebOS 具有以下特点:

- 客户端平台无关,支持“随时随地”的计算和存储,用户可使用位于不同地点的、具有浏览器功能的计算设备进行便捷的移动办公;

- 用户无需为数据建档和备份,由服务器端完成相应功能,可避免由于客户端系统损坏或重装而发生数据丢失;

- 采用虚拟桌面文件系统,能够屏蔽可能位于多台物理机器上的数据,简化使用和管理。

上述特点使得 WebOS 更适合面向瘦客户端用户,其使用模式正得到广泛关注。从 2007 年到 2008 年 7 月,Internet 上可供访问使用的 WebOS 系统从 10 个发展到 20 余个。其中, EyeOS 桌面为 MacOS 风格, Goowy 为基于 Tab 的风格,其它多为 Windows 风格。通过 AJAX, Web Service, J2EE, PHP, Flash 等开发技术,现在 WebOS 已经可以提供许多日常桌面应用,如音频视频播放器、照片编辑工具, E-Mail 客户端、即时信息、日历、文件存储、文件共享、桌面搜索、游戏、Widgets 支持及集成的其它应用等。其中部分 WebOS 还提供开放的应用编程接口,支持进行二次开发,如 YouOS, EyeOS 等。此外,网络上还出现了专业提供 WebOS 的网站,如国外的 Goowy, EyeOS 及国内的 Pc2n。

Internet 已经从简单的文字页面进化到具有丰富应用、交互性很强的新互联网。Google 于 2008 年 9 月推出了网络

浏览器 Chrome, 该浏览器整合了搜索、个性化推荐、JavaScript 引擎 V8 及相对独立的标签页等特性,通过浏览器能够方便地整合 Google Docs, GFS, Gmail 等功能。使得本地机器不用存储大量数据,访问自己需要的数据,数据在 Internet 上处理、最终结果返回给用户,减少了瘦客户机的运行负担和开销。Google Apps 作为 Google 的 WebOS 产品,提供 Start Page, Gmail, GTalk, GVideo, Calendar, Google docs 等功能。另外,微软 Windows Live 对基于网页的应用程序的支持及 Salesforce.com 的按需定制软件服务的发展在一定程度上也将加速 WebOS 这一网络使用模式的推广。

WebOS 是未来网络发展的一种趋势,也是整合网络资源的一种有效的方法。目前,WebOS 还是处于百家争鸣的发展初期,它适应面向网络技术的最新应用需求,目的是为桌面和服务端操作系统提供增值功能并开拓新的使用模式。为使 WebOS 得到更广泛的推广和应用,在桌面客户端,需提高访问服务的易用性、界面的友好性;在服务器端,要求提升服务功能和性能、加强服务可管理性;在客户端和服务器的连接方面,需采用如 AJAX 等技术寻求优化。WebOS 是否会走向标准化还有待分析,但未来 OS 是否提供这样的功能将成为其具备竞争力与否的一个指标。

4 制约 WebOS 发展的主要技术因素

WebOS 从提出至今已经得到了比较迅速的发展,但仍然存在一些技术因素,使其难以被广为接受和使用:

- (1) 处理和传输速率不够快。WebOS 基于浏览器,客户端通常执行 JavaScript 代码。客户端发出请求后,服务器端向客户端发送 JavaScript 代码和有关资源,这些代码和资源在客户端的浏览器中形成桌面窗口应用。但 JavaScript 运行速度较之 Java, C, PHP 等语言而言要慢,因此对于功能强大、组成复杂的应用服务,其性能无法得到保障。需要对多种客户端技术进行性能分析,选择使用对 WebOS 性能提高更加有利的开发技术。另外,多媒体等应用需要浏览器与服务端的高速通信,带宽的限制制约了网络传输的速率,也在一定程度上影响到 WebOS 的用户体验。

- (2) 安全性有待提高。WebOS 基于 HTTP 协议进行客户端和服务端的通信,传输的数据都是明文,其传送的数据和信息容易在网络中被窃取。以 EyeOS 为例,其用户与服务器的报文传输过程中无安全措施,一旦被窃取,就可以冒充该用户发送信息并获得服务。另外,WebOS 支持多用户,用户数据存储在服务器中,供用户随时随地进行访问,这就给数据的完整性和私密性提出了要求。

- (3) 服务器端应用开发环境有待完善和人性化。WebOS 通常用于向用户提供基于网络应用的虚拟桌面环境。由于所用开发语言和使用浏览器等原因,目前 WebOS 在功能上与桌面程序相比差距还较大。虽然不断有 IT 厂商和开源软件开发者将基于桌面的程序改写为基于网页的、适用于 WebOS 的程序,但也仅限于部分应用。因此 WebOS 服务器端应用开发支撑非常重要。目前,开放服务器端 API 接口的 WebOS 不多,且通常采用私有开发接口。如 EyeOS,其服务器端采用 PHP 语言编写。PHP 的编辑、测试软件有 Eclipse, Zend Studio 等,但开发 EyeOS 应用程序时无法使用这些工具调试所编代码,不得不使用手工添加调试代码,进行多次测试才能对

代码功能部分进行错误排查,工作量和开发难度较大。

5 实例:基于 EyeOS 的服务器端功能开发

5.1 EyeOS 原理和组成分析

EyeOS 是基于 Apache, PHP, Ajax, DHTML 开发的开源软件。从 2005 年起, EyeOS 已经升级到 1.8.5, 其代码更新快, 受到了比较广泛的关注。如图 1 所示, EyeOS 为 B/S 结构。

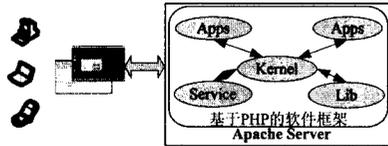


图 1 EyeOS 组成

EyeOS 服务器端采用 PHP 语言开发, 运行于 Apache 服务器中。服务器端由基本服务 (Services)、系统库 (Libs)、内核 (Kernel) 及应用 (Apps) 4 部分组成^[4]。EyeOS 的设计思路沿用了传统操作系统的概念, 但基本服务、系统库、内核都是在传统操作系统之上的用户态程序。

基本服务是由低层的多个公共功能模块组成, 包括安全、XML 管理、日志、用户管理、消息映射、虚拟文件系统、进程管理等。系统库使用上类似系统服务, 只是并非低端功能模块, 是方便应用开发的功能模块, 包括 eyeZip, errorCodes, eyeIPC, eyeCrypt, eyeSockets, il8n 支持等。应用是指 EyeOS 中集成的桌面应用及用户定制应用, 如日历、在线 office、文件管理器等。内核用于定位基本服务和系统库, 并管理这两者与应用的通信。基本服务和系统库为应用提供公共接口。

基本服务具体包括:

- 1) 虚拟文件系统 VFS: 封装了 PHP 文件操作, 支持文件访问权限管理, 提供文件和文件元信息管理, 将系统文件和用户文件分离;
- 2) 用户管理器 UM: 管理用户注册、登录等;
- 3) 消息映射 MMAP: 用于客户、服务器间的消息通信;
- 4) 进程管理 PROC: 每个应用实例作为一个进程, 负责进程创建、显示等;
- 5) XML 管理 eyeX: 管理服务器端返回的 XML 信息在浏览器中创建图形控件;
- 6) 外部服务 Extern: 支持客户端直接通过浏览器 URL 形式获取服务器端文件;
- 7) 日志服务 Log: 管理日志;
- 8) 安全服务 Sec: 提供 disable PHP 某些不安全选项开关等安全配置功能。

系统库包括:

- 1) errorCodes: 确定错误类型并显示错误信息;
- 2) eyeXML: 管理用 XML 形式表示的数组;
- 3) Il8n 国家化: 提供语言支持;
- 4) eyeWidgets: 支持快速图形接口创建。

EyeOS 中应用的存储和结构具有统一的模式。应用程序的目录和文件路径如下:

- eyeOS/apps/: 包含每个应用程序 PHP 代码的目录;
- eyeOS/apps/Application/: 应用程序的主目录;
- eyeOS/apps/Application/apps. eyecode: 初始化和结束的应用程序代码;

• eyeOS/apps/Application/events. eyecode: 动作响应代码;

• eyeOS/extern/apps/Application: 应用程序外部资源目录。

用户和应用通过事件进行交互。例如, 用户点击按钮产生一个事件, 产生的事件以消息形式传送到服务器, 负责处理消息的是 MMAP 服务。每个消息都包含应用程序的地址、校验数和事件的名称。为了传递消息, MMAP 在应用程序目录中寻找以 events. eyecode 命名的文件, 然后试图调用名为 ApplicationName_on_EventName 的函数, 传递消息的内容。

EyeOS 界面美观、操作简单, 支持面向特定用户群开发定制的个性化 eyeOS。目前基于 eyeOS 有多个开源子项目, 包括致力于数据同步的 eyeSync、面向在线儿童教育的 eyeEDU 及实验室信息管理系统 eyeLIME (面向生物领域的数据存储格式、试剂和样本数据管理、支持分子结构显示的 3D 图形等功能)。EyeOS 也存在一些需要改进之处, 例如图标显示不稳定, 响应速度比较慢。

5.2 EyeOS 对 Web Services 的支持

Web Services 定义了应用程序如何在 Internet 上实现互操作, 很大程度上拓展了应用程序的功能, 实现了软件的动态提供, 代表着 Internet 技术的重大发展^[5]。

Internet 上已存在多种 Web Services, 通过支持调用这些服务, 能够使 WebOS 有效利用网络上已有的服务资源。若进一步提供服务器端的 Web Services 开发支持, 将有利于在较短周期内丰富 WebOS 服务器端软件功能, 则对提高 WebOS 服务器端软件的互操作性、加速 WebOS 服务器端功能的丰富化有重要意义。

EyeOS 可通过 PHP 语言的 XML-RPC 和 SOAP 函数库来支持 Web Services 互操作。XML-RPC 是运行在 Internet 之上的远程过程调用协议, 建立在 HTTP 之上, XML-RPC 消息数据主体以 XML 形式表示。与 XML-RPC 相比, SOAP 支持的数据类型更为丰富。在下一节, 将给出 EyeOS 中基于 SOAP 协议的一个 Web Service 的实现。

5.3 基于 SOAP 协议的 Travel 服务的设计与实现

在 EyeOS 中开发应用服务的时候, 需要使用 PHP 代码、PHP 库文件、EyeOS 的系统服务和 EyeOS 系统库。可编写简单、无窗口的程序, 也可编写出复杂的窗口程序, 开发者不必关心窗口动作在客户端是如何实现的。为了探索通过 EyeOS 开发 Web Service 的过程, 以期丰富其服务器端功能, 设计和实现了 Travel 服务, 利用了 Internet 上开放接口的机票查询等 Web Services。

5.3.1 Travel 服务流程设计

为实现旅行、出差管理服务, 本节给出服务流程的设计。首先计划行程, 确定行程后 Travel 服务将调用 Web Service 提供有关出差、旅行的信息。然后用户根据行程查询航班/列车表、天气等信息。如果用户满意天气等客观条件, 则根据自身的需求预定或订购飞机票/火车票, 或更改或取消计划。最后行程和住宿安排成功, 则存储这些信息, 结束此次服务。

5.3.2 Travel 服务的实现

Travel 服务在服务器端实现。客户端将请求发送给服务器端的 Travel 服务, 后者利用系统库、系统服务实现逻辑功能, 并通过 SOAP 协议调用 Internet 上开放接口的 Web Ser-

vices 给用户提供服务,调用的结果返回给用户。Travel 服务的组成及其拓扑如图 2 所示。

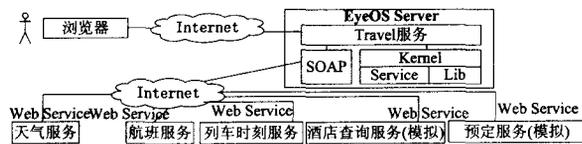


图 2 Travel 服务的实现

用户首次向 Travel 服务发送请求后,Travel 服务启动,在客户端形成窗口形式的管理界面,服务器端继续监听客户请求。当请求为 Web Service 请求时,Travel 服务将调用 Internet 中相应的 Web Service 并等待响应。获得响应后,Travel 服务将相应结果进行解析,并再次处理为用户可接受的格式。随后 Travel 服务将解析处理完的结果发送给用户。若请求的不是 Web Service 请求,Travel 服务将使用 EyeOS 内核调用其系统服务/系统库函数或利用 Travel 服务自身的函数来处理请求,最后将处理结果发送给用户,并且处理信息存储在 EyeOS 服务器中该用户文件夹下的 Travel 目录中。

根据 Travel 服务流程和上述结构设计了 Travel 服务在 EyeOS 中的文件结构和代码结构,并编写了其实现代码,完成对 Internet 上开放 Web Services 的调用和模拟调用。/EyeOS/Apps/Travel/存放 Travel 服务的源代码,主要是 app. eyecode 和 events. eyecode。app. eyecode 中是初始化和结束 Travel 服务的代码,其中 Travel_run 为 Travel 服务初始化并读取相关数据的函数,Travel_end 为结束 Travel 服务的函数。/EyeOS/Users/user's file/files/travel/存放行程计划。

events. eyecode 中实现了 Travel 服务中的 9 个函数:

- getWeather(\$ to):通过 SOAP 协议调用天气预报服务。
- getAirlineInfo(\$ from,\$ to,\$ date):通过 SOAP 协议调用航班信息服务。
- getTrainInfo(\$ from,\$ to):通过 SOAP 协议调用最新列车时刻表服务。
- getHotelinformation(\$ params=""):通过 SOAP 协议调用酒店信息服务的函数(该函数为模拟实现,Internet 中未提供免费的此类服务)。
- arrange(\$ params=""):通过 SOAP 协议调用 Web Service 安排旅行、出差(该函数为模拟实现,Internet 中不能免费使用该服务)。

(上接第 74 页)

[6] Erradi A, Maheshwari P. wsBus; QoS-aware middleware for reliable web services interactions [J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327

[7] Sheth A, et al. Qos for service-oriented middleware[C]// Proceedings of the Conference on Systemics, Cybernetics and Informatics, Orlando, FL, July 2002

[8] Li Y, et al. Self-reconfiguration of service-based systems; A case study for service level agreements and resource optimization [C]// IEEE International Conference on Web Services. 2005; 266-273

[9] Alwagait E, Ghandeharizadeh S. DeW; a dependable web services framework [C]// Proceedings of the 14th International Workshop on Research Issues on Data Engineering; Web Services for E-Commerce and E-Government Applications. 2004;

• Travel_on_addTravel(\$ params="):点击图形界面中 Add Travel 按钮时的响应函数。

• Travel_on_Add():点击 Add 按钮时的响应函数,将填写的信息存储到/EyeOS/Users/user's file/files/travel/目录中,且将调用前 4 个函数返回给 Travel 服务的结果传送给用户。

• Travel_on_Arrange():点击 Arrange 时的响应函数。

• Travel_on_editTravel():点击 Edit Travel 时的响应函数,编辑已计划的旅行、出差。

• Travel_on_delTravel():点击 Delete 时的响应函数,用于删除计划。

EyeOS 中 Travel 服务运行时的截图如图 3 所示。



图 3 运行中的 Travel 服务

结束语 WebOS 随时随地可用的特点使得它将成为未来瘦客户端 Web 应用的一种重要使用模式。本文从 WebOS 的由来、概念和组成入手,分析了 WebOS 的发展现状和趋势,论述了制约其发展的主要技术因素;选择有代表性的开源 EyeOS 进行了深入研究,并对其功能进行了扩充。下一步将重点研究 WebOS 客户端/服务器的高效数据同步机制。

参考文献

[1] Vahdat A, et al. WebOS: Operating System Services for Wide Area Applications[C]// Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing. 1998; 52-63

[2] 武秀川,鞠九滨. WebOS: Internet 操作系统[J]. 计算机科学, 2001, 28(11): 1-5, 12

[3] 邹鹏,阳国贵,等. 操作系统原理与实践[M]. 北京:高等教育出版社, 2008

[4] Norte J C, Fiestas A. EyeOS Developer Manual[EB/OL]. http://www. eyeos. org, 2008

[5] 顾宁,刘家茂,柴晓路,等. Web Services 原理与研发实践[M]. 北京:机械工业出版社, 2006

111-118

[10] Guo H, et al. ANGEL: Optimal Configuration for High Available Service Composition[C]// IEEE International Conference on Web Services (ICWS 2007). 2007; 280-287

[11] Lin K J, et al. Building Accountability Middleware to Support Dependable SOA[J]. IEEE Internet Computing, 2009, 13(2): 16-25

[12] Huang G, Liu X, Mei H. SOAR: towards dependable Service-Oriented Architecture via reflective middleware[J]. Journal of Simulation and Process Modelling, 2007

[13] Zeng L, et al. QoS-aware middleware for web services composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327

[14] Menasce D A. Composing web services: A QoS view[J]. IEEE Internet Computing, 2004, 8(6): 88-90