

# 嵌入式瘦 Web 服务器的设计与研究

崔丽群 张明杰

(辽宁工程技术大学软件学院 葫芦岛 125105)

**摘要** 为了在低端单片机的环境下设计与研究一种嵌入式瘦 Web 服务器,对 TCP/IP 协议栈进行了精简,尤其是在 TCP 精简中提出了一种加速打开机制,避免了三次握手过程,并对 CGI 技术进行了改进。利用 Javascript 将服务器端的部分工作移到客户端处理,利用 XMLHttpRequest 实现服务器和浏览器异步通信,从而能有效地提高服务器的性能。通过性能测试表明,该服务器符合嵌入式环境的要求,在低端平台上实现嵌入式 Web 服务器具有较高的实用价值。

**关键词** 嵌入式瘦 Web 服务器, TCP/IP, 加速打开机制, CGI, Javascript

中图分类号 TP316 文献标识码 A

## Design and Research of Embedded Thin Web Server

CUI Li-qun ZHANG Ming-jie

(College of Software, Liaoning Technical University, Huludao 125105, China)

**Abstract** To design and research of an Embedded Thin Web Server in the environment of low-end microcontroller, streamlining the TCP/IP protocol stack, particularly, putting forward an accelerated open mechanism for TCP streamlining to avoid a three-way handshake process and has improved the technology of CGI. Using the Javascript to make part of the work of the server shifted to the client to process, using XMLHttpRequest implements Asynchronous communication between the server and the browser, so the performance of the server can be improved effectively. Through the test of performance shows that the server can meet the requirements of the embedded environment and this embedded Web server has a high practical value in the low-end platforms.

**Keywords** Embedded thin Web server, TCP/IP, Accelerated open mechanism, Common gateway interface, Javascript

近些年来,随着 Internet 的繁荣发展,在电子信息技术领域中,嵌入式系统与互联网相结合的需求越来越强烈。利用嵌入式 Web 服务器处理嵌入式系统的信息可以获取实时的反馈。特别是进入 3G 时代,嵌入式系统的发展更是蒸蒸日上,它的应用范围无所不在。然而,在嵌入式系统的使用中,将 Web 应用技术融入进来既是当前的一种需要,也会成为人们对嵌入式系统的一个基本要求。因此,在对嵌入式系统设备的研究和开发中,将 Web 应用技术融入嵌入式系统设备有着重大的意义。

目前嵌入式系统大范围的应用主要集中在 8 位或 16 位的低端 MCU 上,在这样的嵌入式设备中实现 Web 服务器的困难在于其资源有限。为了充分利用有限的嵌入式资源,最大程度地降低成本,提高系统的性能,嵌入式 Web 服务器应该实现为瘦 Web 服务器的形式,瘦 Web 服务器是随着 Web 应用环境的改变而提出的。在嵌入式应用当中,Web 服务器不可能很全面,它必须拥有足够小的代码量,且不能影响嵌入式系统的整体性能,但又必须具备一个 Web 服务器应有的特征。本文就是在低端单片机环境下研究具有交互功能的嵌入式瘦 Web 服务器。

## 1 嵌入式瘦 Web 服务器的硬件

### 1.1 处理器的选择

由于低端单片机的运行速度以及嵌入式系统资源的限

制,用普通的单片机实现协议栈,将很难进行其它应用功能的扩展,而用高性能的单片机,其成本较高。在提高性能的同时,为降低成本,可以采用 SX52 微控制器。

SX52 微控制器是由美国 Ubicom 公司研制的高速可配置通信控制器,具备最高可达 100 MHz 的钟频。SX52 没有传统的数据总线和地址总线,而是具备 5 个配置灵活、功能强大的 I/O 端口 (PortA-E),共 40 个 I/O 管脚,每一管脚都可通过软件进行灵活的配置。它采用改进型 Harvard 结构,具备高速的计算能力,可以通过运行软件模块来取代许多传统的硬件设备完成大量的实时功能。

### 1.2 网络接口芯片的选择

采用 RTL8019AS 以太网控制芯片来作为网络接口芯片。RTL8019AS 是一种全双工即插即用的以太网控制器,它在一块芯片上集成了 RTL8019 内核和一个 16kB 的 SDRAM 存储器,实现了基于 Ethernet 协议的 MAC 层的全部功能,内置 16kB 的 SRAM、双 DMA 通道和 FIFO 完成数据包的接收和发送功能。RTL8019AS 有两种数据模式 (16 位和 8 位可供选择),将其 Jp 引脚采取硬跳线的方法强制为高电平,从而选定为 8 位数据模式。

### 1.3 硬件结构

嵌入式瘦 Web 服务器用 SX52 微控制器作为处理器,以太网驱动芯片 RTL8019AS 经网卡隔离器 HR61101G 和 RJ45 接口接入以太网,配有 RS23 和 CAN 总线两个扩展接



指定一个 48bit 的连接计数 cookie。一个主机的 cookie 值从一个全局计数器中获得,每次使用该计数器时加 1。当 server 主机收到 client 的一个 SYN 报文段,读取报文段中携带的 cookie 值,如果该 cookie 值比已经缓存在 client 的 cookie 值大,则该 SYN 是新的,服务器将接收报文段中的任何数据,跳过三次握手过程。

如果接收到的 cookie 比缓存中的 cookie 值小,由于嵌入式瘦 Web 服务器在 TCP 连接时只需处于被动服务的状态,可以将标准的 TCP 状态机中的主动创建连接 SYN\_SENT 状态,主动关闭连接 FIN\_WAIT1、FIN\_WAIT2、CLOSING、TIME\_WAIT 状态都省去,还可以省去 CLOSED 状态,让它一开始就处于 LISTEN 状态来监听客户端的连接请求。它接收到 SYN 数据片时,会发出 SYN+ACK 数据片用以确认来自对方的 SYN,这时的状态为 SYN\_RCVD,再收到对方返回的一个仅含 ACK 的空数据片,则三次握手完成,进入 ESTABLISHED 状态,之后就可以进行 TCP 数据通讯。

## 2.5 零拷贝技术在 TCP/IP 中的应用

为了保证协议栈层与层之间的封装性,TCP/IP 协议栈在实现过程中采取层次拷贝的原则。在嵌入式系统环境下,特别是资源极其有限的低端单片机系统及 SOC 片上的系统,如果在封装和解封装时层层拷贝数据,不仅复杂,而且浪费拷贝时间。TCP/IP 网络系统中,普遍采用“零拷贝”技术来解决这个问题,即在 TCP/IP 协议栈不建立专门用于各层数据传输的缓冲区,协议栈各层之间传递的是指针,只有当数据要被驱动程序发送出去或者被应用程序取走时,才进行真正的数据移动。

最后在实现 TCP/IP 协议栈的过程要注意“内存泄露”问题,要严格遵循“谁分配谁释放”的原则来管理内存,防止系统因可用内存被耗尽而导致嵌入式设备崩溃现象的发生。

## 3 嵌入式瘦 Web 服务器的设计

### 3.1 服务器端 CGI 的改进

目前 Web 技术中生成动态 Web 页面的方法有 CGI (Common Gateway Interface)和服务器脚本,如 JSP、ASP 等。而对于嵌入式 Web 服务器,由于其资源有限、编程语言限制等因素,在嵌入式设备中 Web 方式应用一般是基于 CGI 的程序开发。SSI(Server Side Include)技术也适用于嵌入式 Web 服务器。

针对 CGI 实现过程中系统可维护性差、程序的可读性和扩展性差等问题,需要对 CGI 进行改进,其改进的思想就是在嵌入式瘦 Web 服务器中实现 CGI 程序的过程中引入 SSI 技术,并由 CGI 程序调用。CGI 程序在嵌入式瘦 Web 服务器中以函数的形式实现,CGI 程序可以处理 GET 请求,通过 POST 请求方式提交的 Form 标签对客户端请求进行应答。SSI 在嵌入式 Web 服务器中以 SSI 解析函数的形式实现,取消了标准的 SSI 命令,只对 HTML 页面中的标签进行解析。

当 HTTP 服务器收到来自客户的请求时,调用 CGI 程序。对于动态信息多的页面,由 CGI 程序中的 C 代码直接生成,对于只有少量动态信息的页面或静态页面,则从文件中读取相应的 HTML 模板,交由 SSI 解析函数,SSI 解析函数从文件中获取相应 HTML 模板。值得注意的是在对 HTML 模板文件进行处理之前,先判断其是否包含 SSI 标签。如果

含有 SSI 标签,则需对该 HTML 模板进行解析与处理,将 SSI 标签替换为相应的数据。若该 HTML 模板中不含 SSI 标签,则直接将该 HTML 模板作为静态网页输出。

在嵌入式服务器端进行处理的过程中,为了满足应用程序和驱动程序互相配合的需要,可以将使用驱动程序的应用程序的代码直接加到 CGI 程序的某一段中。这样做的缺点是网页代码和网页实现代码交叉混合在一起不利于系统。为了解决以上缺点,可以将 CGI、驱动程序、使用驱动程序的应用程序分开,CGI 和该应用程序之间通过 linux 的 shell 来传递参数的方法,使得 CGI 不与驱动程序直接联系,提高了程序的独立性和开发效率。

### 3.2 服务器端页面文档管理

在服务器端设计页面较多,需要用到 HTML 模板,在网页生成阶段也会用到一些文档、图片等,可以设计一种页面文档管理结构来管理这些文档。

在程序中建立一个标准的文件来存储静态网页和动态网页,每个文件包括文件名、文件内容指针、文件大小、文件最大尺寸、内容更新函数指针、配置文件更新函数指针、表单数据解析函数指针。在系统初始化的时候,一次性分配一个比所有页面最大尺寸之和略大的连续内存空间。为了提高嵌入式 Web 服务器的运行效率,将各个页面的最大尺寸依次与该页面对应的文件内容指针相关联。当需要对该系统的某个文件进行修改而导致页面内容发生变化时,只需修改该文件内容指针指向的那个文件的内容即可。同样,在保存文件配置数据时,也只需通过配置文件更新函数指针找到相应的配置文件更新函数进行操作即可。这样有效地避免了因修改文件内容而要频繁申请内存的操作,也避免了为存储页面内容而频繁进行 Flash 读写操作。

### 3.3 客户端的验证

本文通过在浏览器中执行嵌入在 HTML 页面里的 Javascript 程序对用户输入端信息做客户端的前期校验,对输入诸如密码等信息进行验证。如果信息通过验证,则会通过与嵌入式 Web 服务器的交互来实现下一页面的浏览,但如果信息不能通过验证,则浏览器端会弹出输入的信息不符等对话框,不再增加服务器端的任务量。因此,客户端验证可以减少服务器端的资源占用情况,减少服务器端的任务量,使得用户能更加迅速地浏览页面。

嵌入网页的 Javascript 程序在浏览器中被执行使得验证功能成为现实。当 Javascript 代码认为输入的数据不能通过验证时,便用该代码中的 alert()方法弹出用以提醒错误的警告对话框。

### 3.4 客户端处理

客户端处理就是将服务器端发送的变量和数据传给 Javascript 程序,充分利用浏览器端 Javascript 程序的计算能力。本文利用 Javascript 程序对客户端进行验证以及将服务器端的任务部分转移至客户端来实现。通过 XMLHttpRequest 对象,可以实现与服务器只在数据层面上进行交换,无需每次都刷新页面,并且接收到服务器返回的数据还可以交给在浏览器上执行的 Javascript 程序来处理,从而实现服务器和浏览器的异步通信。Javascript 可以嵌入到网页中,并通过内置于浏览器中的 Javascript 解释器对其进行解释执行,

(下转第 275 页)

OL]. <http://www.ibm.com/developerworks/cn/linux/l-migrate2cloud-1/>, 2012-07-20

- [6] Khajeh-Hosseini A, Greenwood D, Sommerville I. Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS [C]//2010 IEEE 3rd International Conference on Cloud Computing. Miami, USA: IEEE Computer Society, 2010: 450-457

- [7] Hajjat M, Sun Xin, Sung Yu-wei E, et al. Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud [C]// SIGCOMM 2010. Association for Computing Machinery, 2010
- [8] 李强, 郝沁汾, 肖利民, 等. 云计算中虚拟机放置的自适应管理与多目标优化[J]. 计算机学报, 2011, 34(12)

(上接第 238 页)

通过该技术,利用客户端资源来实现数据的计算等复杂操作。XMLHttpRequest 组件可以利用 Javascript 程序来创建其对象及其函数的应用来实现客户端对服务器端数据的访问及计算更新。

客户端处理具体步骤:首先将计算函数嵌入 Javascript 脚本;然后将该函数定义成能实现链接的 onClick 事件处理程序;最后当用户点击界面按钮能链接该事件时,查找相应的计算函数来计算参数,该参数即为服务器传至浏览器的数值。

### 3.5 客户端缓存

XMLHttpRequest 组件与服务器端的信息交互需要经常与服务器建立连接的原因在于该组件向服务区提交请求的方法为 HTTP 法。而一旦需要交互、更新的数据过多,就需要大量多次地连接服务器,使得本来资源有限的服务器负载增大。为了避免这种缺陷,本文在客户端中增加一种缓存机制。该缓存机制在网页中是不可视的,它的任务在于减少 XMLHttpRequest 发送请求时与服务器进行的连接次数。该缓存可以将多个请求整合,由此来进行集体对服务器的连接,使得连接次数减少。服务器端也会将请求的资源内容进行整合封装,从而形成一个应答报文与浏览器端进行交互。这样也能减少两者的连接次数,减少服务器端的负载。而封装的信息由缓存来解析,并由相应的回调函数进行处理。

## 4 性能测试

在 SX52+RTL8019AS 平台上对该嵌入式瘦 Web 服务器从收到请求建立连接到响应结束断开连接的时间进行了测试。在实验中将本文设计的嵌入式瘦 Web 服务器与使用 Applet 的 Web 服务器进行比较,从而找出其性能优势及不足。其中比较的前提是访问的是同一个网页且完成的是同一个功能。其性能比较如表 1 所列。

表 1 性能比较表

静态页面	数据流量(k)	测试次数	平均响应时间(ms)
本文 Web	0.83	20	2.72
使用 Applet	1.22	20	2.85
动态页面	数据流量(k)	测试次数	平均响应时间(ms)
本文 Web	3.25	20	22.98
使用 Applet	6.43	20	24.46

通过对性能测试表的分析,表明该嵌入式瘦 Web 服务器具有比较快的服务响应速度,可以正常执行设计要求的各项功能。AJAX 技术的使用明显降低了服务器的响应时间,并减少了 Web 流量。AJAX 技术的使用,一方面可以利用客户端闲置的处理能力承担一部分服务器工作,减轻服务器和带宽的负担;另一方面可以降低页面重载的频率,减少带宽消耗,优化性能,可以得到更好的用户体验。

上述结果证明了使用 AJAX+CGI 的思想构造嵌入式瘦 Web 服务器的方法是可行的。该思想可以充分利用客户端资源,明显降低服务器端负载,提高在低端嵌入式设备中嵌入式 Web 服务器的性能。

**结束语** 本文主要是在低端单片机的环境下实现具有交互功能的嵌入式瘦 Web 服务器。通过此设计,进一步了解了嵌入式系统网络接入和嵌入式瘦 Web 服务器的基本原理、组成、结构和实现过程。该嵌入式瘦 Web 服务器能够满足低端嵌入式设备接入 Internet 并通过 Web 服务器进行监控的要求,且能够基本符合嵌入式系统开发过程中低成本、高可用性的要求。但是在系统中嵌入式瘦 Web 服务器对信息以及数据的传递上,需要加强安全保密工作,以提高数据传输的安全性。

## 参考文献

- [1] Su Ying-qiang, Zhang Wei-qiang. Design of wireless embedded thin web server based on zigbee [C]// 2010 2nd International Conference on Future Computer and Communication (ICFCC). 2010(1): 525-528
- [2] Liu Li-xia. Research on technology of embedded web server application [C]// 2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME). 2010: 187-189
- [3] Liu Jian-hong, Chen Jing, Tai Yu-chin, et al. Supporting Audio Streaming in Application Cloud for Embedded Systems [C]// 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICISS). 2012 IEEE 14th International Conference on High Performance Computing and Communication, 2012: 1800-1805
- [4] Liu Jian-hong, Jing Chen, Chuan Tai-yi, et al. ACES-Application Cloud for Embedded Systems [C]// 2011 IEEE/IPSJ 11th International Symposium on Applications and the Internet. 2011: 145-151
- [5] Kovatsch M, Mayer S, Ostermaier B. Moving Application Logic from the Firmware to the Cloud: Towards the Thin Server Architecture for the Internet of Things [C]// 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. 2012: 751-756
- [6] Limpraptono F Y, Ratna A A P, Sudibyo H. Remote laboratories multiuser based on embedded web server [C]// 2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV). 2012: 1-7
- [7] Limpraptono F Y, Sudibyo H, Ratna A A P, et al. The design of embedded web server for remote laboratories microcontroller system experiment [C]// TENCON 2011-2011 IEEE Region 10 Conference. 2011: 1198-1202
- [8] Karia D C, Adajania V, Agrawal M, et al. Embedded Web server application based automation and monitoring system [C]// 2011 International Conference on DSignal Processing, Communication, Computing and Networking Technologies (ICSCCN). 2011: 634-637
- [9] Amiri R, Elkeelany O. An embedded TCP / IP hard core for Smart Grid information and communication networks [C]// 2012 44th Southeastern Symposium on System Theory (SSST). 2012: 185-189