

Research and Implementation of Innovative WebSocket Push Technology Based on GoAhead Server

Jianbin Wang

Electromechanical Control, China Electronics Technology Group Corporation's 10th Research Institute, Chengdu, Sichuan, 610036, China

Abstract

The GoAhead server, as a web server for embedded platforms, is widely used in various embedded platforms such as Linux, VxWorks, and Reworks. Due to the fact that traditional technologies do not support the server's ability to actively push messages to clients and the function of asynchronous message processing, the client is unable to display the server status in real time. This paper aims to study the innovative WebSocket pushing technology based on the GoAhead server. By deeply analyzing the architecture of the GoAhead server and combining the advantages of the WebSocket protocol, the pushing function and asynchronous message processing function based on the GoAhead server are finally successfully implemented. Through testing, this research enables the changes in the server status to be quickly reflected on the client side, greatly improving the real-time performance of data interaction.

Keywords

GoAhead; websocket; HTTP; Asynchronous Message Processing; Active Push

基于 GoAhead 服务器的创新型 websocket 推送技术与实现

王建斌

中国电子科技集团公司第十研究所, 中国·四川·成都 610036

摘要

GoAhead服务器作为一款为嵌入式平台web服务器, 广泛应用于Linux、VxWorks和Reworks等多种嵌入式平台。由于传统技术不支持服务器主动向客户端推送消息和异步消息处理功能, 导致客户端不能实时显示服务器状态。本文旨在研究基于GoAhead服务器的创新型websocket推送技术, 通过深入剖析GoAhead服务器架构, 结合websocket协议的优势, 实现了基于GoAhead服务器的推送功能与异步消息处理功能, 经测试, 该研究使得服务器状态变化能迅速反馈至客户端, 极大提升了数据交互的实时性。

关键词

GoAhead; websocket; HTTP; 异步消息处理; 主动推送

1 引言

随着计算硬件、软件技术、互联网技术以及嵌入式技术的飞速发展, 嵌入式设备被广泛应用到工业控制系统和通信设备中^[1]。但是嵌入式设备在运行过程中, 往往需要进行状态监控, 以便及时发现问题并采取相应措施。由于嵌入式设备通常缺乏人机交互界面, 远程监控成为了必然需求^[2]。一般嵌入式设备通过以太网总线进行远程监视, 以太网总线

的嵌入式设备远程监视系统分为 B/S 架构和 C/S 架构^[3]。C/S 架构虽然在数据处理和交互方面具有一定优势, 但它需要开发人员额外开发远程监控客户端软件, 且不同的客户端软件对部署的计算机硬件有要求, 这增加了开发和使用的复杂性^[3]。相比之下, B/S 架构只需在浏览器中输入设备的 IP 地址, 就能实现远程监控, 极大地简化了操作流程, 因此受到了广泛关注^[4]。

为了解决这一问题, 业界提出了多种解决方案。其中, websocket 技术以其独特的全双工通信特性脱颖而出。它不仅支持浏览器与服务器之间的“请求-响应”通信, 还支持服务器主动向浏览器推送数据。当嵌入式设备状态发生变化时, 服务器可主动向浏览器推送最新工作状态, 浏览器接收到数据后更新页面, 从而实现了设备状态的实时监控, 提升了客户端实时监测的能力^[7]。

【基金项目】红果园军工项目(项目编号: M17GY500160)。

【作者简介】王建斌(1985-), 男, 硕士, 工程师, 从事机电控制研究。

2 websocket 协议

websocket 协议是伴随 HTML5 发布的一种创新型网络通信协议,协议包括握手协议和数据传输协议,可以传输基于文本的消息和二进制数据,实现了浏览器与服务器之间的全双工通信^[5]。websocket 协议也发挥着重要作用。金融市场行情瞬息万变,通过 websocket 协议,服务器可以实时将最新的价格、成交量等数据推送给客户端,投资者能够及时获取这些信息,做出准确的投资决策。在物联网设备的状态监控与远程控制场景中,websocket 协议同样不可或缺。例如,智能家居系统中的各种设备,如智能摄像头、智能门锁、智能家电等,通过 websocket 协议与服务器建立连接,服务器可以实时监控设备的状态,并根据用户的指令远程控制设备,实现智能化的家居管理,为用户提供更加便捷、舒适的生活体验^[6]。

websocket 协议的通信过程主要包括握手和数据传输两个关键阶段^[7],在数据传输阶段,websocket 协议采用了独特的数据传输帧格式,以确保数据的准确、高效传输。数据传输帧格式包含了多个关键字段,每个字段都有其特定的含义和作用,通过这种数据传输帧格式,websocket 协议能够在保证数据传输可靠性的同时,提高传输效率,满足不同应用场景对数据传输的要求。

3 基于 GoAhead 的 websocket 模块设计

3.1 模块总体架构

基于 GoAhead 的服务器框架,在设计 websocket 模块时,需要兼容现有框架功能同时也要满足 websocket 协议。基于 GoAhead 的 websocket 模块设计主要握手处理子模块、数据处理子模块和主动推送子模块这三个核心部分协同构成,各子模块分工明确、紧密配合,共同实现高效的服务器推送功能和异步消息处理功能。

3.2 子模块功能设计

3.2.1 握手处理子模块

握手处理子模块主要完成与浏览器的 websocket 连接请求处理功能。握手处理流程如 5 所示。首先由浏览器客户端向服务器发起 TCP 连接请求,服务器 Accept 浏览器连接后,在本地创建一个对应套接字,通过套接字接收客户端浏览器发送 websocket 连接请求。当 websocket 模块接收到浏览器的握手连接 HTTP 请求数据包后,请求包如图 3 所示。首先检测连接路径是否已经在 websocket 路径管理集合中注册,如果检测失败,则回复客户端消息代码为 404 非法连接;如果检测成功,则将该套接字回调接收处理 HTTP 协议接口更新为处理 websocket 数据接口,后续接收到数据按照 websocket 协议进行处理,处理流程如图 9 所示,同时将该套接字以“IP+ 端口”为索引,添加到连接管理集合中,为客户端主动向浏览器推送消息提供接口。完成对 websocket

连接请求处理后,通过调用 `websGetVar(wp," HTTP_SEC_WEBSOCKET_KEY", 0)` 获取客户端连接 key 值,利用 sha 算法计算回复 key 值,向服务器上报消息代码 101(切换协议代码),通知客户端浏览器从接收 HTTP 协议数据切换到接收 websocket 协议数据,回复报文如图 4 所示。GoAhead 服务器框架为了提高套接字的利用率,对每一个 TCP 连接套接字进行 15s 是否有数据超时检测,如果在 15s 之内没有接收到任务数据,则执行关闭套接字操作。由于客户端浏览器与 websocket 模块套接字连接成功后,创建一个本地的 websocket 套接字,通过 websocket 套接字与客户端浏览器建立长时间通信,交互数据为触发式数据并非周期数据,因此在 websocket 模块确认与浏览器客户端连接成功后,通知 GoAhead 框架关闭对应 websocket 套接字的超时检测。

3.2.2 数据处理子模块

数据处理子模块主要完成接收浏览器下发数据,完成数据的分发和处理。首先从 websocket 套接字上接收 4 字节包头数据,按照协议帧格式解析负载数据长度。由于 websocket 协议采用的 TCP 数据流传输,因此在接收负载数据时,首先需要解析其长度,然后根据长度接收对应的负载数据。当接收到的负载数据与包头长度相等时,则证明接收到完整一包 websocket 数据,对负载数据进行 mask 转换后,根据应用程序注册的回调接口,处理浏览器客户端下发数据。

3.2.3 主动推送子模块

主动推送子模块主要完成将应用将软件数据推送到所有已经建立连接的浏览器客户端,主动推送消息流程由应用软件发起,不依赖于浏览器客户端请求。主动推送子模块通过遍历连接管理集合中所有已经连接的浏览器客户端,既可以完成文本数据推送,同时也可以完成二进制数据推送。

3.2.4 性能测试

基于以上 GoAhead 服务器 websocket 模块设计,本文搭建了一个模拟实际应用场景的测试环境,对 B/S 架构与 C/S 架构进行对比测试。

本次性能测试主要关注延迟时间这一关键指标,延迟时间是衡量系统实时性和响应速度的重要参数,它直接影响着用户体验和系统的实用性。在实际应用中,如实时监控系统、即时通讯软件等,低延迟对于及时获取信息和进行交互至关重要。

客户端采用 Perntium Dual CPU 主频 1.6GHz、内存未 4Gmb,服务器 PowerPC2020 主频 1.6GHz、内存 2Gmb,浏览器为谷歌浏览器。

在测试过程中,为了保证测试结果的可靠性和准确性,对每个数据长度进行了多次测试,避免因单次测试的偶然性因素导致结果偏差。对测试数据进行了严格的记录和整理,对不同长度数据延迟如表 3 所示。

表3 性能测试结果

| 长度 (Byte) | B/S 延迟 (ms) | C/S 延迟 (ms) |
|-----------|-------------|-------------|
| 4096 | < 1 | < 1 |
| 8192 | 1.5 | 1.3 |
| 10240 | 2.4 | 2.1 |
| 20480 | 3.5 | 3.2 |
| 40960 | 4.9 | 4.6 |
| 长度 (Byte) | B/S 延迟 (ms) | C/S 延迟 (ms) |
| 4096 | < 1 | < 1 |

从表3测试结果可以看出,对于相同长度的数据,B/S架构与C/S架构的延迟相差低于1毫秒。在数据长度为4096字节时,B/S架构和C/S架构的延迟均小于1毫秒,这表明在处理小数据量时,两种架构都能快速响应,满足实时性要求。随着数据长度的增加,如在8192字节时,B/S架构延迟为1.5毫秒,C/S架构延迟为1.3毫秒;在10240字节时,B/S架构延迟为2.4毫秒,C/S架构延迟为2.1毫秒。虽然B/S架构的延迟略高于C/S架构,但两者的差距非常小,在实际应用中,这种微小的延迟差异对于用户界面软件几乎没有影响。

这种结果的产生与两种架构的通信原理和数据处理方式密切相关。C/S架构中客户端与服务器直接建立连接,数据传输路径相对较短,且客户端可以分担一部分数据处理任务,因此在延迟方面具有一定的优势。而B/S架构虽然需要通过浏览器进行通信,增加了一定的网络开销和处理环节,但通过优化的websocket协议和高效的服务器实现,有效地降低了延迟,使其在性能上与C/S架构相当。在实际应用场景中,如实时监控系统、即时通讯软件等,这种低延迟的性能表现能够确保数据的及时传输和交互,为用户提供流畅的使用体验。对于实时监控系统,设备状态的变化能够及时推送给客户端,让监控人员能够迅速做出反应;在即时通讯软件中,消息能够快速传递,实现即时沟通的效果。

4 结论

本文研究了客户端浏览器与服务器之间异步响应和服务器主动向浏览器客户端推送数据的技术。利用websocket协议,设计实现了基于GoAhead服务器的websocket模块,实现了浏览器与服务器之间异步响应和服务器主动推送数据功能,解决了GoAhead服务器的B/S系统架构中,服务器状态变化后不能及时通知客户端更新的问题,提升了B/S系统架构中浏览器客户端界面监控的实时性。在实时监控系统中,服务器能够及时将设备状态变化推送给客户端,使监控人员能够迅速做出反应;在即时通讯软件中,消息能够快速传递,实现即时沟通的效果。这一成果为嵌入式设备的远程监控提供了更高效、更可靠的解决方案,提升了B/S系统架构中浏览器客户端界面监控的实时性,具有重要的理论意义和实际应用价值。

参考文献

- [1] SALUNKEPG, SAYYED A M Design of embedded web server based on NIOS-II soft core processor[C/OL]//2016 Internation Conference on Eletrical, Electronics and Optimization Techniques(ICEEOT) Chen gna:IEEE,2016:488-492[2019-0420]
- [2] 孙留存,苏卫江,库斯达,等.基于嵌入式的智能远程视频监控系统设计[J].自动化与仪器仪表,2024,(08):163-167.
- [3] 程耀,谢华,万思宇,等.基于嵌入式Linux的双通道DVR系统设计与实现[J].自动化与仪器仪表,2022,(11):179-183.
- [4] 刘雪梅,王卫军.基于嵌入式的步进电机远程控制技术[J].自动化与仪器仪表,2021,(07):76-79.
- [5] 章立宗,孙洲,王华慧,等.基于分布式区块链的储能数据交互处理模型设计[J].电子设计工程,2023,31(15):70-74.
- [6] 张帝,周宾斌,孙畅子.基于嵌入式Linux的智能点餐系统的设计与实现[J/OL].企业科技与发展,1-5[2025-04-16].
- [7] 胡洋洋.基于WebSocket的服务器推送技术的研究与实现[D].南京邮电大学,2018.