B.原文的翻译

使用WebSocket开发基于Web的控制程序

**摘要**

WebSocket 允许基于Web的应用程序与Web服务器之间进行异步全双工通信。WebSocket开始于HTML5标准化的部分但是现在已经脱离HTML5而单独开发。使用WebSocket使得开发独立于平台的表示层应用加速器和光束控制软件变得容易。此外，Web浏览器是唯一需要安装在客户端计算机的应用程序。WebSocket应用使用简单的基于文本的消息的通信，所以适用于基于消息的控制系统如MADOCA。使用WebSocket技术已经实现了一个简单的MADOCA控制系统和一个简单的电机控制中的应用。使用谷歌浏览器（13版）在Debian / Linux和Windows 7，Opera浏览器（11版）在Debian / Linux和Safari（版本5.0.3）在Mac OS X作为客户，电机可以控制使用WebSocket的基础应用。并且已经在同步辐射衍射实验衍射仪控制应用中使用了。

**介绍**

基于Web的应用程序相对于使用图形用户界面开发的定制应用程序相比有许多优点：1）平台的独立性；2）易于开发；3）易于分发最新版本。基于这些优势，已经有许多尝试引入基于网络的应用程序的加速器和控制器。

平台独立性来自于这样一个事实：大多数的主流Web浏览器不仅可以在传统的个人计算机或工作站的操作系统如微软的Windows， Mac OS X，大多数的Linux发行版，Solaris和FreeBSD，还可以运行在移动设备的操作系统，如Android和iOS。这给了我们许多可供选择的操作控制台和不同类型的加速器和物理实验操作。使用移动设备，控制员可以访问后端的设备，并且可以很容易地解决设备问题，如果需要的话。

有很多可供开发基于Web的应用程序的工具，包括独立的设计和开发工具，如微软的Visual Studio和Eclipse，和浏览器内置的工具。使用这些工具开发应用程序比使用其他语言和工具包更容易。

一个基于网络的应用程序的每个执行单元都加载于一个网络服务器。维护所有运行应用程序的最新版本是很容易的，因为你需要更换网络服务器上的网络应用程序，你不需要分发和要求你的应用程序用户安装新版本。

尽管有这些优势，基于Web的应用程序有一个很大的缺点，即一个Web应用程序（基于HTML的应用程序）不能进行全双工通信与服务器应用程序由于HTTP通信开始从客户端。这意味着一个服务器不能从客户端发送任何信息到客户端。

为解决这一问题，Comet技术中引入了“长轮询”机制。Comet服务器不会立即响应客户端的请求（通常使用XHR）。它等待一个服务器事件，服务器则在事件的信息上对客户端进行响应。实际上，Comet服务器响应会控制在一定时间（通常为30秒）因为浏览器检测HTTP会话超时。

这一机制运作得很好。然而，它需要对每次长轮询请求开始一个新的HTTP通信会话。这是一个沉重的负载，并且它不可能在Comet服务器在长轮询间隔期间发送信息给客户端。这意味着，使用Comet很难实现一个“真正的”控制应用程序。

WebSocket协议给出了给真正的双向通信能力。WebSocket目前是独立的HTML5标准，但是是一个HTML5重要组成部分。

WebSocket在基于Web的应用程序中非常容易处理，可以在程序中用JavaScript方便地操作。WebSocket可以处理文本和二进制数据。然而，它更容易在JavaScript应用程序处理文本数据。这意味着它很容易开发了一个基于WebSocket的以文本消息为基础的控制系统，如MADOCA或STARS的应用。一旦你使用WebSocket作为控制网关，你就可以开发基于JavaScript的应用。因此，我们构建了一个针对MADOCA的网关（WebSocket服务器）和几个基于Web的应用程序以进行了测试，这些在下面的章节中描述。

**服务器端实现**

一个作为madoca网关的使用WebSocket服务器的控制系统进来开发了出来。该服务器需要处理WebSocket协议的协商阶段。该协议已修订的最新版本是hybi-10但大多数Web浏览器上实现hybi-00；因此，我们在hybi-00 WebSocket服务器的基础上实现的。WebSocket服务器可以与HTTP服务器使用相同的端口，即并存，80（或443）。在这种情况下，Web服务器必须处理WebSocket谈判阶段，而我们的WebSocket服务器运行在10101端口，因为服务的连续性。控制在Spring-8 X射线光束线的光学元件，要求服务器在端口10101。

该服务器是一个简单的socket服务器和一个小的修改，我们可以调整服务器的两个简单的socket服务和WebSocket服务。服务器从第一接收到的字符串中决定所需的服务。对于简单的Socket服务的MADOCA格式信息将收到与WebSocket服务定义的格式在hybi-00或hybi-09将收到的，这是一个HTTP请求，即相同，“GET /demo HTTP / 1.1”或“GET / WS HTTP / 1.1”。在建立与客户端的连接，WebSocket服务器从客户端接收MADOCA格式发送到MADOCA为基础的控制系统。WebSocket服务器并接收来自控制系统的MADOCA并将它们发送回客户端。

**客户端程序**

客户端应用程序是用JavaScript写的。在Javascirpt，你可以使用“WS”对象打开WebSocket，使用ws.send()发送消息，使用ws.onmessage() = function(event) { var message = event.data; … }来接受消息。请注意，接收程序会被异步调用。所以你需要一个事件驱动的处理程序来处理它。如上图所示，WebSocket接口可以很容易地处理。

**支持的浏览器**

以下浏览器在PC机上进行了测试，发现工作正常：谷歌浏览器13在Windows 上，Debian GNU Linux和Ubuntu / 11.04；在Mac OS X Safari；Firefox 4在Ubuntu 11.04；和Opera 11.50在Debian GNU Linux和Ubuntu 11.04 /。Opera和Firefox，WebSocket功能默认被关闭，所以你需要在“关于：配置”中打开此功能。在移动设备上的Web浏览器，Opera移动11.10在Android 2.3.3（你需要启用WebSocket）和Safari浏览器的iPad（第一代）可为WebSocket客户端。图4显示了一个衍射仪控制运行在“三星Galaxy Tab”是一个基于Android系统的平板电脑配备了7液晶屏。

**结论**

在本文中，Web应用程序可以使用WebSocket协议搭建。我要强调的是，没有任何附加功能是执行WebSocket应用所需要的，只需要一个基于HTML5的Web浏览器。基于HTML5的Web应用程序提供了自由的平台，所以你可以使用基于Web的应用不仅在中央控制室还与各种类型的设备。这意味着网络应用程序可以用于本地调整或设备故障。基于网络的应用程序也对远程实验很有用。您可以很容易地维护最新的应用程序在远程用户的计算机上运行，因为该应用程序是从网络服务器加载的。