

网页浏览器内核的比较研究*

牛 津¹ 杨 涛² 王 林²

(¹ 北京信息控制研究所 北京 100037 ² 北京神舟航天软件技术有限公司 北京 100094)

摘要:JavaScript引擎和排版引擎是浏览器内核的关键部件。研究了各JavaScript引擎的内部结构及技术特点,分析了各排版引擎的特点及应用,介绍了引擎的测试工具,并对不同JavaScript引擎和排版引擎的性能进行了对比测试。

关键词:浏览器 JavaScript引擎 排版引擎 性能测试 内核 比较

Comparison and Research of Web Browser Internals

NIU Jin¹, YANG Tao², WANG lin²

(¹ Beijing Institute of Information and Control, Beijing, 100037

² Beijing Shenzhou Aerospace Software Technology Co., Ltd, Beijing, 100094)

Abstract: JavaScript engine and layout engine form the basis of it. This paper explores the internal structure and implementation techniques of JavaScript engines, and describes features of various layout engines with their applications. After introducing test tools of JavaScript engines and layout engines, this paper compares the performances of JavaScript engines and layout engines.

Keywords: Browser, JavaScript engine, layout engine, performance test, internal, comparison

网页浏览器作为个人用户最经常使用的客户端程序,用于显示 World Wide Web 或 Intranet 等的文字、图像及其他信息,是个人用户访问互联网的入口。另外,各种基于 WEB 的办公系统、业务系统等商业应用程序也已在企业中广泛部署。目前常见的网页浏览器包括微软的 Internet Explorer、Mozilla 的 Firefox、苹果的 Safari 以及 Google 的 Chrome 等。

网页浏览器的主要工作是发送用户请求,接收服务器端响应,将返回的 HTML 页面展现给用户。随着 Web 2.0 的推进,以及作为核心支撑技术的 Ajax 的普及,对 JavaScript 的处理也成为决定 WEB 页面特效及用户操作体验的重要因素,这主要取决于浏览器内核中 JavaScript 引擎的表现。另外,随着用户的浏览内容的多样化,页面的渲染速度成为了评价浏览器好坏的重要因素,这是由浏览器内核中的排版引擎决定的。本文通过对主流浏览器的 JavaScript 引擎及排版引擎的比较研究,得出不同 JavaScript 引擎和排版引擎的特点,并使用测试工具对引擎的性能进行了评测。

1 JavaScript 引擎

JavaScript 引擎的功能是解释 JavaScript 源代码并执行。目前流行的 JavaScript 引擎如表 1 所示。

本文于 2008-12-12 收到。

* 基金项目:科技部科技支撑计划 2006BAF01A04 基于数字样机的新一代运载火箭综合集成和协同应用研究。

1.1 SpiderMonkey 和 TraceMonkey

(1) SpiderMonkey。SpiderMonkey 是历史上第一个 JavaScript 引擎,以 C 语言实现,由 Netscape 通信公司的 Brendan Eich 编写,目前作为开源项目由 Mozilla 基金会维护。

SpiderMonkey 包括编译器/反编译器、解释器、垃圾收集器,以及标准类(standard class)几个部分。首先由 `jsparse.c` 中的 `parser` 将源代码解析为语法树,然后调用 `jsemit.c` 中的 `js_EmitTree` 函数生成字节码,最后由 `jsinterp.c` 中的 `js_Interpreter` 函数将字节码解释执行。

SpiderMonkey 采用 `generation-mark-and-sweep` 的垃圾收集器,对新产生的对象优先收集。该收集器仅在垃圾收集对象所占空间达到上限且有新的存储空间分配请求时才收集。

(2) TraceMonkey。TraceMonkey 引擎是 SpiderMonkey 的升级版,增加了 JIT 编译器。该 JIT 编译器在 Tamarin 引擎(ActionScript 引擎,用于 Adobe Flash)的 `nanojit` 编译器基础上开发,提高了处理 JavaScript 的效率,使其更接近直接编译的效率。与常见的 JIT 编译器不同,它并不编译整个方法,而是追踪^[1](tracing)内部频繁执行的操作(主要是循环操作),然后对必要部分(如循环体内的操作)进行编译,追踪过程所需的信息由追踪树提供。目前该 JIT 所生成的机器码只支持 `x86` 和 `arm` 系列 CPU。

在多项测试中,使用了追踪 JIT 的 TraceMonkey 性能相比 SpiderMonkey 有相当的提升,如图 1 所示。

1.2 JavaScriptCore、SquirrelFish 和 SquirrelFish Extreme

(1) JavaScriptCore。JavaScriptCore 由 KJS(KDE 的 JavaScript engine)及 PCRE 正则表达式库发展而来,在 WebKit3.1 版本中使用。它采用基于语法树的解释器,以及 `generational` 的垃圾收集器。

JavaScriptCore 的解释器工作时需要对语法树进行遍历,该过程需要频繁访问语法树上并不产生作用的结点,如语句块两端的“`{}`”、“`}`”等,而且遍历过程中需要频繁传递数据,另外,对树上的结点每次访问都需要一次虚函数调用和返回,这些操作相当费时。

(2) SquirrelFish。SquirrelFish 引擎是 JavaScriptCore 引擎的升级版,在 WebKit3.1 以上版本中使用。为了消除 JavaScript 语法树遍历的缺点,它使用字节码解释器(bytecode interpreter)代替了原来的基于语法树的解释器,即将源代码转换为字节码,再对字节码解释执行。它还使用了直接链接代码(direct threaded code)技术来优化解释器的性能,即在解释执行字节

表 1 流行 JavaScript 引擎的对比

名称	实现语言	主要应用范围
SpiderMonkey, TraceMonkey	C	Firefox
JavaScriptCore, SquirrelFish, SquirrelFishExtreme	C	Safari
V8	C++	Chrome
Windows Scripting Host	未知	IE

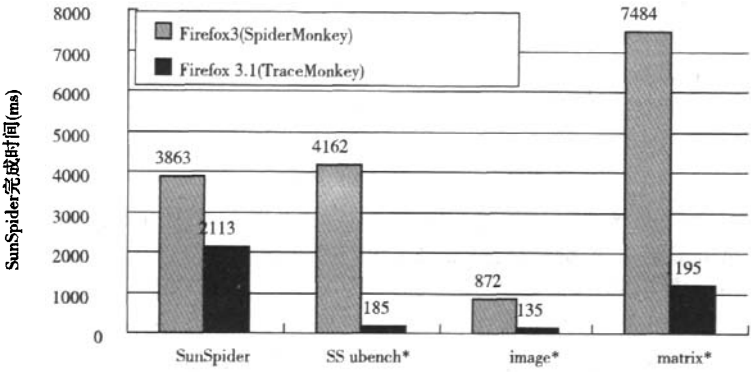


图 1 SpiderMonkey 和 TraceMonkey 的性能对比

*注:SS ubench 是 WebKit 用于测试 SquirrelFish 的测试工具,image 项使用的是 Mozilla 发布的用于测试图像处理能力的 demo,matrix 项使用的是 Sylvester 的矩阵运算库。

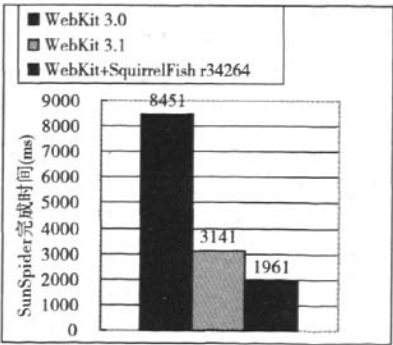


图 2 SquirrelFish 与 JavaScriptCore 的性能对比

码之前把字节码数组转换为直接包含字节码对应解释程序入口地址的数组^[2],提高了解释执行的效率。

图 2 是 WebKit 官方公布的性能数据,可以看出,使用 SquirrelFish 引擎的 WEBKIT 相比使用 JavaScript-Core 引擎的 WEBKIT3.1 速度提高了约 60%。

另外,与其他浏览器的 JavaScript 引擎相比,其速度优势明显。经测试,SquirrelFish 的速度是 SpiderMonkey(用于 Firefox 3.0 RC1 中)速度的 1.46 倍,是带追踪的 Tamarin 的 1.9 倍,是 vanilla 版 Tamarin(用于 FLASH9 中)的 1.8 倍。

(3) SquirrelFish Extreme。SquirrelFish Extreme 引擎于 2008 年 9 月发布,是对 SquirrelFish 的改进版本。其改进之处有以下四点:①对字节码的优化。包括字节码指令集的改进,如添加组合指令、窥孔优化(peephole optimizations)、更快的操作常量和专门用于执行常用操作的机器码。②多态内联缓存(Polymorphic Inline Cache)。对象的属性访问是 JavaScript 处理性能的瓶颈,对于具有相同属性、且属性顺序相同的对象,用一个结构 ID 来标识其对象,使得结构相似的对象对属性进行访问时,很可能执行的是同一段代码,从而使效率得到提高。③上下文链接的实时编译器(Context Threaded JIT)。将部分字节码转换为机器码——将复杂的操作码转换为函数调用,将简单的操作码、或常用的复杂操作码直接内联为机器码流,使硬件的分支转移能力得以发挥^[3]。④正则表达式实时编译器。将实时编译技术应用到正则表达式的处理过程中,使其处理速度提高了 5 倍,并提供了原来仅在 Perl、Python 或 Ruby 中支持的文本处理功能。

图 3 是 WebKit 官方公布的性能数据,可以看出,使用 SquirrelFishExtreme 的 WEBKIT 引擎相比使用 SquirrelFish 的 WEBKIT 速度提高了约 110%。

1.3 V8 引擎

V8 引擎是 Google 为其 Chrome 浏览器专门开发的 JavaScript 引擎,使用 C++ 语言实现,支持 ECMA-262 第三版的标准,可运行于 Windows XP、Vista、Mac OS X 10.5, Linux 等系统上。

V8 引擎的独特之处是其动态机器码生成系统,它直接将 JavaScript 源代码编译为机器码以提高效率,该过程不生成字节码。另外,V8 使用动态创建隐含类(hidden class)的方法实现属性的快速访问,该方法的思想与 Self 语言使用 map 结构的原理类似^[4],即当某类的对象第一次被创建时,将建立一个隐含类,当对象被添加了新的属性后,将生成新的隐含类。这种处理方法对动态语言主要有两个好处:一是属性的访问不需要额外的查找(dictionary lookup),二是 V8 能够使用经典的基于类的优化——内联缓存(inline caching)。

V8 采用 stop-the-world、generational 的垃圾收集器,而且,它始终了解内存中对象和指针的位置(accurate),避免将对象误认为是指针而导致的内存泄露问题。

1.4 Windows Scripting Host

在 IE 中,对 JavaScript 的处理是通过 WSH(Windows Scripting Host)来完成的。WSH 是 Windows 中自带的脚本运行环境,以 COM^[5] 技术为基础,支持 VBScript, JScript 两种语言。

WSH 包含了 Script Engine 和 Scripting Host 两部分,其中 Script Engine 为脚本的解释器,负责对 JavaScript 的解释和执行。当所执行的动作需要 Scripting Host 所提供的服务如读写文件时,需向 Scripting Host 提出要求,由它进行处理。

WSH 使用 non-generational 的垃圾收集器,结构较为简单,性能相比其他引擎的 generational 的垃圾收集器略差。

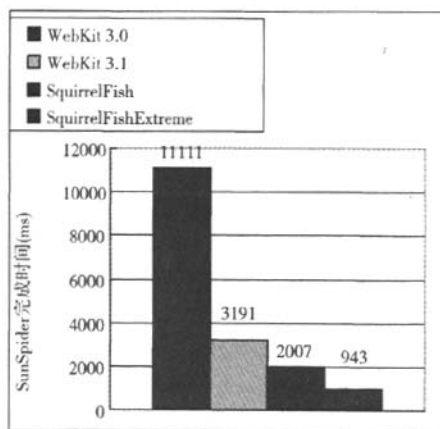


图3 SquirrelFishExtreme 与 SquirrelFish 的性能对比

2 排版引擎

排版引擎又叫渲染引擎(rendering engine),在浏览器软件中负责展现内容(如 HTML、XML、图片、applets 等)和整理信息(如 CSS、HTML 标签等),并将内容格式化输出。除了网页浏览器之外,电子邮件客户端以及其他需要编辑、显示网络内容的应用程序都需要排版引擎。

目前流行的排版引擎如表 2 所示。

表 2 流行排版引擎的对比

名称	实现语言	主要应用范围
Trident	未知	IE
Gecko	C++	Firefox
KHTML	C++	Safari, Chrome

2.1 Trident

Trident 又名 MSHTML,第一版随 IE 4.0 的发布而推出,是目前使用最为广泛的排版引擎。在 Trident 中,微软首次提出了著名的 COM(组件对象模型)技术,以增强 Windows 中软件组件的通信和协作。

除 IE 之外,使用 Trident 引擎的浏览器有:Avant Browser、Maxthon、GreenBrowser 等。另外,Google Talk, AOL Instant Messenger 6.x 以及微软的系列软件 MSN Messenger、Outlook、Windows Media Player 等均使用了 Trident 引擎进行内容的渲染。IE8 beta 版中将包含最新的 Trident VI,而目前已发布的 IE 7 正式版中使用的是 Trident V,它所支持的特性有:HTML 4.01、HTML 5 子集、CSS Level 1(不完全支持 CSS 2 和 3)、JScript 5.6(JavaScript 1.4)、DOM Level 1(不完全支持 Level 2)。

2.2 Gecko

Gecko 是一套开源的网页排版引擎,被 Firefox 浏览器以及 Netscape 6 以上版本使用,流行程度仅次于 Trident。其组成包括:文档解析器(解析 HTML 和 XML)、样式系统(处理 CSS 等)、图形库、平台相关的图形、渲染组件(支持 Win32、X、Mac)等。

除了 Firefox 之外,使用 Gecko 排版引擎的浏览器有:Camino、Flock、SeaMonkey、K-Meleon、Epiphany 等等。最新的 Gecko 1.9 使用了 Cairo 2D 图形库以取代原有的 GFX 图形库,它能够使用显卡的 GPU 来渲染 WEB 页面的 2D 图像,提高了渲染速度,并大幅减轻了 CPU 的负担^[5]。它支持的特性有:HTML 4.0、CSS Level 1(不完全支持 CSS 2 和 3)、JavaScript 1.8、DOM 1、2(不完全支持 DOM 3)、XML 1.0、XHTML 1.1、XSLT 和 XPath、MathML、XForms(需要官方扩展插件)、RDF。

2.3 KHTML

KHTML 由 KDE 开发,以 LGPL 方式授权,在 2002 年被苹果采纳,用来开发 Safari 浏览器,并将包装后的 KHTML 命名为 WebCore,作为 WebKit 浏览器框架的排版引擎。KHTML 拥有速度快捷的优点,但语法容错度比 Gecko 引擎小。

除了 Safari 之外,使用 KHTML 的浏览器还有:Konqueror、Google Chrome、Nokia Series60 的浏览器等。最新的 KHTML 支持的特性包括:HTML 4.01、CSS 1 和 CSS 2.1、CSS 3 选择符(selector)及部分其他功能、ECMA-262(JavaScript 1.5)、PNG、JPEG、GIF 图形格式、DOM 1、DOM2 及部分的 DOM 3、SVG。

3 性能测试

3.1 JavaScript 引擎测试

3.1.1 测试工具

(1)SunSpider。使用最为广泛的 JavaScript 引擎测试工具,由 WebKit 团队于 2007 年 12 月发布,用于测试不同浏览器、浏览器不同版本的 JavaScript 性能。测试集与实际的 JavaScript 应用较为接近,包括浮点数运算、数组访问、位运算、控制流、加密、日期类、数学运算、正则表达式、字符串处理等 9 项测试,测试结果以通过所有 9 项测试的总时间表示。测试中不包含 DOM 操作测试。

(2)Chrome Benchmark。由 Chrome 团队开发,包含五个测试环节。该测试对递归性能较为看重,不包含

DOM 操作测试。

(3)Dromaeo V2。由 jQuery 作者和 Mozilla 的 JavaScript 权威 John Resig 开发,测试从内置函数到主流函数库等等实际浏览时会遇到的运算问题,包括大量有关 DOM 操作的测试。

本文选用 SunSpider 作为测试工具,以获得贴近实际应用的测试结果。

3.1.2 测试结果

对 Firefox、Safari、Chrome 浏览器分别进行测试,机器配置为 Core 2 Duo CPU, 2GB RAM,测试环境为 Windows Vista 32-bit,禁用所有的浏览器插件。

从图4可以看出,使用 V8 引擎的 Chrome 以微弱的优势胜过使用 SquirrelFishExtreme 的 Safari 4.0 排名第一,带有追踪功能的 Firefox 3.1 beta 1 也表现优异,排名第三。由于 IE 的测试时间过长(总耗时 27124 毫秒,仅 string 单项测试即耗时 18296.8 毫秒),未在上图予以标示。

3.2 排版引擎测试

3.2.1 测试工具

ACID:ACID 是由网页标准计划小组(Web Standards Project, WaSP)设计的一份网页浏览器与网页各种标准兼容性的测试网页,主要针对 DOM 和 JavaScript 问题,通过可以度量的标准进行测试。最新版本 Acid 3 包括 100 个测试子集,分为 6 组(bucket),测试结果以图形表示:6 个矩形分别代表 6 组的测试成绩,矩形的颜色表示每组内通过的测试子集的个数。

本文选用最新的 Acid3 作为测试工具。

3.2.2 测试结果

对以 Trident, Gecko, KHTML 为核心的浏览器分别进行测试,机器配置为 Core 2 Duo CPU, 2GB RAM,测试环境为 Windows Vista 32-bit,禁用所有的浏览器插件。

从图5可以看出,基于 Trident 的 IE 对新的标准如 CSS2 的支持较差,即使是使用最新发布的 IE 8,其 ACID 得分也仅有 21 分。基于 Gecko 1.9.1 的 Firefox 3.1 Beta2,其得分较基于 Gecko 1.9 的 Firefox 3.0.4 有较大提升,排名第二。而配备了 WEBCORE (KHTML) 核心的 Safari,其最新版本 4.0 Beta 甚至已经完全通过了 Acid3 的测试,得分达到了 100 分,同样以 WEBCORE 为核心的 Chrome 对比标准的支持也较好,得分仅次于 Firefox 3.1b2 和 Safari 4.0 Beta。

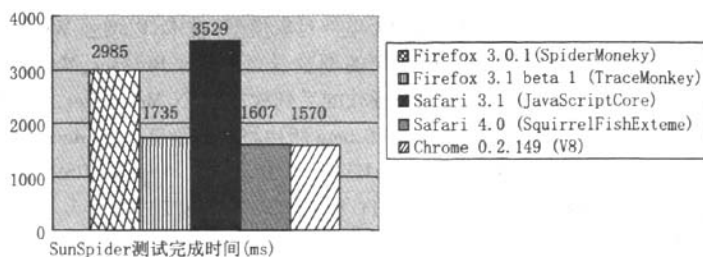


图4 不同JavaScript引擎的性能对比

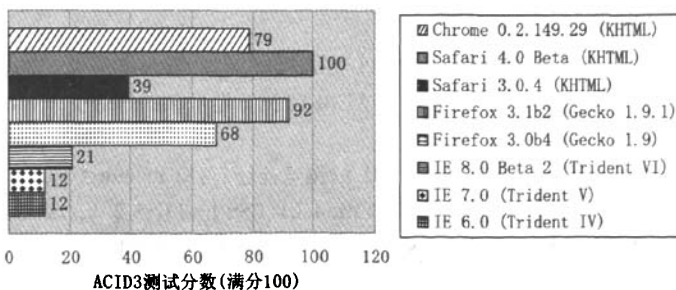


图5 不同排版引擎的性能对比

4 结束语

本文介绍了当前主流浏览器的 JavaScript 引擎和排版引擎的实现原理及技术特点,并使用评测工具对各浏览器 JavaScript 引擎和排版引擎的进行了评测,比较了各 JavaScript 引擎的性能差异及排版引擎对网络标

准的支持程度。在桌面应用 WEB 化及企业的网络应用日益增多的趋势下,浏览器作为客户端的入口程序,其重要性不言而喻,比较研究各浏览器的内核技术,分析浏览器内核的性能,对开发和部署基于 WEB 的应用程序,有一定的参考意义,对于浏览器及插件的开发人员,亦有重要的实用价值。

参 考 文 献

- 1 Andreas Gal, Michael Franz. Incremental Dynamic Code Generation with Trace Trees Technical Report No. 06 - 16, Donald Bren School of Information and Computer Science, University of California, Irvine, 2006.
- 2 马嘉,周明天,陈虹. 一种基于 ARM7 的嵌入式 Java 虚拟机性能优化技术研究. 计算机应用研究, 2007, 24: 97 ~ 100
- 3 Marc Berndt, Benjamin Vitale, Mathew Zaleski, Angela Demke Brown. Context Threading: A flexible and efficient dispatch technique for virtual machine interpreters. Code Generation and Optimization, 2005. CGO 2005. International Symposium on, 2005: 15 ~ 26
- 4 Craig Chambers, David Ungar, Elgin Lee. An Efficient Implementation of Self, a Dynamically - Typed Object - Oriented Language Based on Prototypes. OOPSLA 89 Conference Proceedings, 1989: 49 ~ 70
- 5 梁忠杰, 思敏, 李婷. COM 技术和动态链接库技术的应用研究. 微计算机应用, 2006, 6
- 6 张健浪. 显卡加速网页渲染. 个人电脑, 2007, 3:

作者简介

牛津,男,(1985 -),硕士研究生,主要研究方向为产品数据管理。

杨涛,男,(1974 -),硕士研究生,工程师,主要研究领域:产品数据管理。

王林,男,(1972 -),硕士研究生,工程师,主要研究领域:研发过程管理、数据库应用。

网页浏览器内核的比较研究

作者: 牛津, 杨涛, 王林, NIU Jin, YANG Tao, WANG lin

作者单位: 牛津, NIU Jin(北京信息控制研究所, 北京, 100037), 杨涛, 王林, YANG Tao, WANG lin(北京神舟航天软件技术有限公司, 北京, 100094)

刊名: 微计算机应用 

英文刊名: MICROCOMPUTER APPLICATIONS

年, 卷(期): 2009, 30(3)

被引用次数: 0次

参考文献(6条)

1. Andreas Gal, Michael Franz [Incremental Dynamic Code Generation with Trace Trees \[Technical Report No. 06-16\]](#) 2006
2. 马嘉, 周明天, 陈虹 [一种基于ARM7的嵌入式Java虚拟机性能优化技术研究\[期刊论文\]-计算机应用研究](#) 2007
3. Marc Berndt, Benjamin Vitale, Mathew Zaleski, Angela Demke Brown [Context Threading: A flexible and efficient dispatch technique for virtual machine interpreters](#) 2005
4. Craig Chambers, David Ungar, Elgin Lee [An Efficient Implementation of Self, a Dynamically-Typed Object-Oriented Language Based on Prototypes](#) 1989
5. 梁忠杰, 思敏, 李婷 [COM技术和动态链接库技术的应用研究\[期刊论文\]-微计算机应用](#) 2006(06)
6. 张健浪 [显卡加速网页渲染\[期刊论文\]-个人电脑](#) 2007(03)

相似文献(10条)

1. 学位论文 王永光 [JavaScript引擎及其相关技术的研究](#) 2009

浏览器是人们上网的一个重要工具。近年来, 随着我国移动通讯业的发展, 手机终端功能的增强, 手机上网的人数一直不断地翻倍增长。但是, 国内的手机浏览器产品与国外的产品相比存在着较大的技术差距。因此, 研发一个具有自主知识产权的优秀的手機浏览器内核的浏览器有十分重要的现实意义。

JavaScript是应用在浏览器里面的一种脚本语言。它能够实现一种动态的和可交互的表达能力。JavaScript引擎是浏览器内核的一个关键组成部分。JavaScript引擎技术是实现浏览器的重要技术。因此, JavaScript引擎技术的研究对实现浏览器内核有巨大的作用。

本课题来源于一个具有自主知识产权的手机浏览器项目的JavaScript引擎的第一阶段研究工作。该项目在深入研究、吸收和消化开源Webkit的基础上, 对Webkit进行移植、修改和优化。因此, JavaScript引擎的研究也就以Webkit的JavaScript引擎JavaScriptCore为研究对象。第一阶段研究的内容是分析JavaScriptCore的体系结构与各组件的主要功能。并在此基础上, 重点研究虚拟机组件与外壳组件。

虚拟机是JavaScriptCore的核心组件之一。它的性能直接影响着JavaScript的执行速度。本文在详细分析虚拟机现有的技术特点基础上, 借鉴多态内联缓存技术的思想, 提出了一个优化对象属性查找的方案。

外壳组件是一个实现浏览器客户端对象的对外接口组件。对它的功能进行了分析, 然后着重研究它的设计, 最后指出存在的设计问题, 提出一个优化方案。

本文的最后分别对提出的两个优化方案进行了简单测试, 证明这两个方案是可行的。

2. 学位论文 刘刚 [基于MiniGUI的嵌入式浏览器的研究与开发](#) 2004

本文首先对嵌入式浏览器的技术、市场现状及其发展前景做了一个简要的介绍和分析, 然后阐述了浏览器设计的基本原理以及Monqueror嵌入式浏览器的整体构架, 并以konqueror-embedded为研究和开发的原型, 开发出一款基于MiniGUI的符合HTML4. 0和W3C标准的嵌入式浏览器。接下来重点讨论了浏览器中的一些关键模块, DOM(Document Object Model)模块详细介绍了DOM标准, 并采用设计模式的观点对整体代码在结构上做了详细说明, 网络模块主要介绍了ghttp库的使用以及cookie的实现, 解析模块则在语法和词法分析的算法上做了详细说明, 最后是对浏览器中的一些关键技术介绍, JavaScript部分主要介绍了如何利用JavaScript引擎SpiderMonkey在浏览器中支持JavaScript, Flash部分则简要介绍了根据本课题的实际情况对Flash支持的折衷方案。

3. 期刊论文 刘春梅, 孙鹏, 胡琳琳 [基于JavaScript引擎的终端动态数据处理方法](#) -微计算机应用2010, 31(4)

随着嵌入式系统的发展, 嵌入式终端有大量的动态数据需要处理。在以浏览器作为用户界面的嵌入式系统中, 由于终端动态数据具有分散化和本地化的特点, 如何使用独立于应用程序之外的浏览器处理终端数据成为一个难点问题。针对这一问题, 提出一种利用浏览器的JavaScript引擎处理终端动态数据的方法, 实现了浏览器对终端各业务数据的处理。本方法实现了动态数据获取与显示的分离, 降低了代码间的耦合度, 增加了用户界面风格的灵活性和多样性。

4. 学位论文 张宏伟 [基于嵌入式浏览器的JS引擎移植的研究与应用开发](#) 2009

近年来, 随着嵌入式技术的飞速发展, 嵌入式浏览器逐渐成为最重要的嵌入式软件之一。嵌入式设备具有多样性, 硬件配置低, 存储空间有限等特点, 所以良好的运行性能对嵌入式浏览器至关重要, 而JavaScript引擎的性能逐渐成为制约浏览器性能的瓶颈。

本文在研究嵌入式浏览器特性和相关技术的基础上, 讨论了Linux平台下嵌入式浏览器JavaScript引擎移植的技术和整体方案设计。首先, 本文分析了某嵌入式浏览器的代码和运行性能, 并针对其性能缺陷介绍了改进的办法, 进而提出替换JavaScript引擎的设计方案。其次, 在设计方案中, 结合当前浏览器的流行框架, 详细介绍了浏览器中BOM(Browser Object Model, 浏览器对象模型), DOM(Document Object Model)和JS(JavaScript)引擎的绑定技术, 并选取代表性很强的对象加以设计和实现。然后, 在浏览器框架结构下, 对JS引擎的核心工作流程(如, 引擎初始化, 对象实现, 对象注册等), 移植过程中所涉及的关键技术(如, 对象实现, DOM绑定, 内存管理等), 关键数据结构, 及接口加以介绍和重点阐述。最后, 对优化前后的浏览器进行了分析和对比。数据显示, 改进后的浏览器性能有了大幅的提升。

本文在研究JS引擎移植的基础上, 完成了在嵌入式浏览器中JS引擎的移植和开发, 使其在实际应用中性能有了大幅的提升。随着嵌入式研究的深入, 相关的应用将会更加广泛, 如增加邮件客户端, 完善对XML的支持等等。

5. 学位论文 [刘刚 基于MiniGUI的嵌入式Web浏览器的研究与开发](#) 2004

在嵌入式系统上开发Web浏览器正逐渐成为一个技术热点,只要是可以接入网络的设备几乎都可以用到Web浏览器,但由于浏览器涉及到一个广泛的技术集合,开发浏览器是一项非常巨大的工程,具有一定的商业风险,所以目前市面上的嵌入式浏览器绝大部分都是国外厂商的,真正国内拥有完全自主知识产权的嵌入式浏览器很少.本文首先对嵌入式浏览器的技术、市场现状及其发展前景做了一个简要的介绍和分析,然后阐述了浏览器设计的基本原理以及Monqueror嵌入式浏览器的整体构架,并以konqueror-embedded为研究和开发的原型,开发出一款基于MiniGUI的符合HTML4.0和W3C标准的嵌入式浏览器.接下来重点讨论了浏览器中的一些关键模块,DOM模块详细介绍了DOM标准,并采用设计模式的观点对整块代码在结构上做了详细说明,网络模块主要介绍了ghttp库的使用以及cookie的实现,解析模块则在语法和词法分析的算法上做了详细说明,最后是对浏览器中的一些关键技术的介绍,JavaScript部分主要介绍了如何利用JavaScript引擎SpiderMonkey在浏览器中支持JavaScript,Flash部分则简要介绍了根据本课题的实际情况对Flash支持的折衷方案.这篇论文对浏览器的基本原理和总体结构做了简要介绍,重点介绍了其中的一些关键模块和关键技术的实现,对我国开发自主知识产权的嵌入式浏览器具有很大的意义.

6. 学位论文 [赵文博 一种基于Rhino的Web功能测试工具的设计与实现](#) 2009

当前主流的Web功能测试工具通常是采用录制回放/关键词驱动的方式进行的,它们一般使用测试脚本来驱动浏览器进行测试.这些测试工具在测试用例较多而且需要频繁运行时,测试速度比较慢的缺点就成为了一个重要的问题.而对于自动化的Web功能测试,尤其是回归测试来说,浏览器的很多功能不是必要的,比如网页展示(Rendering)、用户界面等.

本文设计和实现了一种Web功能测试系统,能够显著提高测试运行速度.本系统通过模拟传统Web功能测试中的浏览器,只提供功能测试需要的功能,去掉页面展示和用户界面等功能,从而达到提高运行速度的目的.为了模拟浏览器进行工作,本文实现了网络交互、HTML/CSS解析、cookie等功能.对于Web应用程序来说,在页面中使用JavaScript是很普遍的,特别是在AJAX网站中JavaScript被大量使用.为了能够执行这些网页中的脚本,本系统中嵌入了一种名为Rhino的开源JavaScript引擎,并以此为基础实现了模拟浏览器中的JavaScript运行的功能.为了让测试人员能够使用本系统进行测试,本文设计并实现了接口来控制浏览行为以及获取页面元素的状态信息.另外,由于现有的Web功能测试工具,例如Selenium已经在广泛应用,所以本系统还实现了运行Selenium录制的测试脚本的功能.

本文实现了170个测试用例运行实验,结果显示系统能够正确模拟浏览器运行,且比驱动浏览器进行测试的工具(Selenium)的运行时间缩短38.6%.此外,本文还使用系统访问了130多万个网页,并获取页面元素的事件绑定信息,验证了系统的稳定性.实验证明,本系统能够提高Web功能测试测试的效率,缩短项目周期.

7. 学位论文 [李英各 嵌入式浏览器中JavaScript的设计与实现](#) 2004

近年来,移动通信产业发展迅猛,已经进入了向即将到来的3G时代过渡的重要阶段.同时,OMA组织的WAP2.0标准规范不断成熟,完善,为无线移动网络和Internet的互连互通扫除了技术障碍.为了支持网页浏览,图像,视频,游戏等高级应用,代表新一代移动通信终端发展趋势的智能手机脱颖而出,嵌入式浏览器也逐渐变成智能手机的标准配置.目前阻碍移动通信产业进一步发展的瓶颈之一就是内容服务相对缺乏,所以当前的嵌入式浏览器不仅要能访问移动互联网的wap1.x网页,还要能够访问互联网中浩瀚的网络资源.鉴于JavaScript在开动态网页中的重要地位,绝大部分互联网网页中都嵌入了大量的JavaScript代码,这就要求嵌入式浏览器支持JavaScript,以增强浏览器的访问能力.本课题的目的就是设计并实现一种面向嵌入式浏览器的JavaScript插件,以支持网页中的JavaScript.考虑到嵌入式环境的特点,我们确定设计的总目标是“代码尺寸小,响应时间短,代码易移植,可剪裁,可灵活配置,准确且高度稳定”.本课题创造性的把SpiderMonkey这一著名的JavaScript引擎应用到了嵌入式领域,在充分分析,研究,消化,吸收其技术的前提下,遵照WAP2.0标准规范,设计并实现了一个基于课题合作公司的嵌入式浏览器DeltaBrowser上面的可商用的JavaScript的解释执行环境.根据在一款真实手机环境下的测试结果,DeltaBrowser使用该插件后不仅可以支持测试网页中的大部分JavaScript代码,并且解决了JavaScript执行过程中消耗内存过多的问题,而且在响应速度上也能满足实际需求.根据目前掌握的资料,国内在这一领域研究不多,稳定性的商用产品还未见上市.

8. 学位论文 [陆晓春 基于服务器端Push的Web框架设计与实现](#) 2007

Web 2.0的应用日益广泛,Ajax在其中起到了非常重要的作用.它提供与服务器异步通讯的能力,让用户从请求/响应的循环中解脱出来,从而带来了更为自然的浏览体验.但在实际开发中,不少应用对实时性提出了更高的要求,在这样的需求下一种新的技术诞生了.

Comet建立在Ajax的基础之上,但对Ajax的现有实现进行了创新.它改变了浏览器从服务器端Pull数据的惯例,转而由服务器直接Push数据给浏览器.双向沟通建立在服务器和客户端之间——服务器端产生的异步事件可以直接发送给客户端;而由事件驱动,客户端可以即时响应服务端的请求.B/S构架向基于C/S结构的程序跨近了一大步,因而可以实现之前无法做到的应用.

本文详细论述了这一技术,以实时性为目的,设计并实现一个基于J2EE的框架,以验证此理论的可行性.设计上以实时流协议为基础,确立了控制通道和数据通道的概念,制定了一套自己的指令与协议.开发包括服务器端和客户端两方面:遵从指定的协议,服务器控制并维持与客户端的连接并以HTTP流传送大数据——从功能以及扩展性、可维护性出发,服务器端划分成会话管理、事件分发、数据源和事件钩子四个模块;同时在客户端实现一个JavaScript引擎,发送指令、接收流数据并分发事件.最后本文列举了几个应用实例,包括将Push技术与SOA结合,设计了一个实时汇率发布的应用,以及将其推广到现有的BBS系统上,对浙大的飘渺水云间Web提出了更新,从实践上论证模型框架的可靠性与准确性;并分析了框架的应用前景,指出目前存在的不足与问题,为下一步的研究和实现奠定了坚实的基础.

9. 期刊论文 [张涛,黄强,毛磊雅,高兴, ZHANG Tao, HUANG Qiang, MAO Lei-ya, GAO Xing 一个基于JSON的对象序列化](#)

[算法 - 计算机工程与应用](#)2007, 43 (15)

目前基于Ajax技术的Web开发主要采用XML进行数据交换,然而XML是一种结构化的文档,需要服务器和客户端都对其进行手工解析,将会占用更多的系统资源,因此采用XML进行数据交换会导致性能低下、兼容性不够、灵敏度低的问题.JSON(JavaScript Object Notation)是一种轻量级的数据交换格式,易于被支持JavaScript的浏览器所解析.提出了一种基于JSON的对象序列化算法,该算法通过分析JSON文法并建立对象导航图,透明地将Java对象序列化或JSON表达式,使客户端能够很好地利用JavaScript引擎来解析JSON响应,有效地解决了解析XML所造成的缺陷.

10. 学位论文 [周林 在嵌入式浏览器中的JavaScript和DOM的支持](#) 2004

本文首先深入研究了JavaScript的特点及如何在嵌入式浏览器中支持JavaScript.JavaScript是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能脚本语言.

其次,本文还探讨了在嵌入式浏览器中如何支持DOM.DOM(Document Object Model)文档对象模型是W3C制定的一个标准API,但因为DOM需要在实际进行处理前对整个HTML或XML文档进行分析,而把整个HTML或XML文档转换而成的树放到内存中需要占据很大的内存空间,特别是当文档很大的时候,所以不太容易在资源相对有限的嵌入式系统中实现.文中通过对W3C DOM规范针对嵌入式系统的特点做出响应的裁剪,在JavaScript引擎中实现JavaScript与DOM绑定的方法,来使嵌入式浏览器实现了对DOM的支持.

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjsjyy200903007.aspx

授权使用: 北京邮电大学(byab), 授权号: 3254c825-343f-477e-9129-9e4e00ab5b26

下载时间: 2010年12月16日