·计算机工程不与应用·

多核平台嵌入式浏览器并行机制的研究与设计

桑 楠，赵 丽，郭文生

(电子科技大学嵌入式实时计算实验室 成都 611731)

【摘要】为了提高嵌入式浏览器在多核处理器上的显示速度，提出了一种针对多核平台的嵌入式浏览器多线程划分并行设计方法。该方法将嵌入式浏览器内核划分为多个线程，以用户界面作为主线程，资源加载、解析、排版布局、图形绘制模块划分为子线程，使嵌入式浏览器以多线程方式在多核处理器上并行执行。该方法打破了传统浏览器的串行过程，克服了传统浏览器的用户响应效果差、多核处理器的CPU利用率低等问题。经实验测试，验证了该方法的可行性，并且在内存不超过64 MB的情况下，网页显示速度提高了18%～40%。

关 键 词 嵌入式浏览器; 排版布局; 资源加载; 多核; 并行设计
中图分类号 TP311 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.03.014

Research and Design of Parallel Method for Multi-Core Embedded Browser

SANG Nan, ZHAO Li, and GUO Wen-sheng

(Laboratory of Embedded Real-Time Computing, University of Electronic Science & Technology of China Chengdu 611731)

Abstract A parallel method with multi-threading for multi-core embedded browser is introduced in order to improve the speed of browser in multi-core processors. By putting the user interface into the master thread, and loading resources, parsing, layout and painting into the slave thread, this method parallels the serial processes for Web pages. This method can solve the problem of the browser's slow response and low CPU utilization of multicore processors. The feasibility of the method is verified and the speed of rendering the web pages is improved by 18%～40% with memory less than 64 MB memory according to the testing cases.

Key words embedded browser; layout; loading resources; multi-core; parallel method

近年来，随着嵌入式设备，尤其是移动设备的发展，嵌入式设备拥有更加强大的处理能力[1]。嵌入式浏览器技术越来越多地应用在机顶盒、手机、家用电器中，具有巨大的应用前景[2]。现有的嵌入式浏览器都是在单核处理器平台上运行，如微软的IE浏览器，苹果的safari浏览器，还有火狐浏览器等[3]。在多核处理器上运行原来的串行浏览器，会出现多核处理器得不到充分利用、浏览器在性能上仍然得不到很大提高的问题。目前，Google的chrome浏览器已经做了多进程的优化[4]，但是该浏览器只是在外壳上做了多进程优化，而内核仍然是采用的串行浏览器内核。

为了适应越来越普遍应用的多核处理器平台，需要重新设计嵌入式浏览器架构，充分利用多核处理器的优势，从而提升嵌入式浏览器的性能和用户体验。为此，本文设计并实现了一个基于WebKit引擎的多核嵌入式浏览器，真正从内核划分多个线程，提高了在多核处理器上的性能。该浏览器具有运行速度快，占用内存小等优点。

1 嵌入式浏览器概述

1.1 嵌入式浏览器介绍

随着嵌入式设备和移动互联网的发展，嵌入式浏览器已经成为主要的嵌入式软件。作为用户与网络交互的首要入口，嵌入式浏览器出现了很多产品，如微软伴随其最新智能手机操作系统Windows Phone 7.5的IE mobile 9，Android系统的Android浏览器，iphone自带的safari浏览器等。

1.2 串行嵌入式浏览器架构分析

WebKit[5]是一个开源的浏览器引擎，源码结构

收稿日期：2013－01－05；修回日期：2013－05－02
基金项目：国家核高基重大专项项目(2012ZX01033-001-001)
作者简介：桑楠(1964－)，男，教授，主要从事嵌入式实时高可信技术、实时软件工程与实时中间件技术、实时/嵌入式系统应用技术方面的研究.

清晰、渲染速度极快。WebKit采用分层设计，主要分为WebKit接口模块、WebCore[6]内核模块和JavascriptCore[7]三大模块，WebCore包括网络加载模块、HTML解析模块、DOM模块、CSS解析模块、排版计算模块以及渲染呈现模块，负责从发起网页加载到最终显示的大部分工作；JavascriptCore[7]完成JavaScript脚本的解析和执行。具体架构如图1所示。

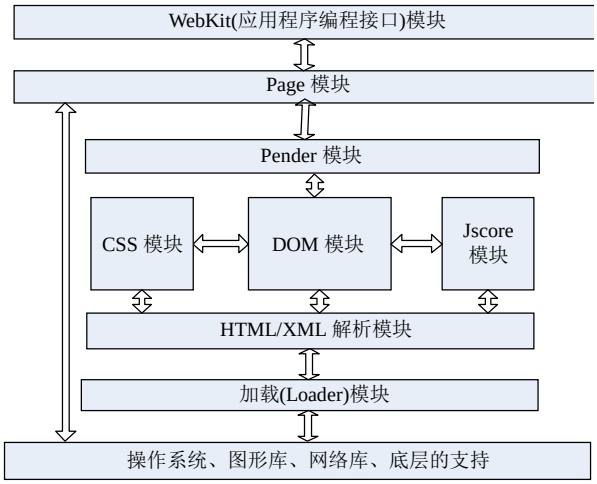

图 1　WebKit 架构

浏览器从用户输入网址开始，由load模块进行资源下载。下载完的数据由浏览器解析模块HTML/XML模块进行网页解析，遇到css代码由css模块解析成对应样式结构保存为render模块使用，遇到js代码由jscore模块进行js解析并执行，网页解析完毕后生成对应的dom树以及包括节点信息和样式信息的render树。render树经过排版布局计算后生成虚拟页面布局，然后调用图形库将页面显示出来。

2　多核概述

随着CPU制造技术的不断进步，功耗问题越来越限制了单核处理器不断提高处理器性能的发展途径，其解决方案之一就是使用多核处理器。多核处理器，是指在一个处理器上集成两个或两个以上的处理引擎，即拥有多个可同时计算的核心，通过并行计算的方式降低功耗、提升性能。

由于多线程技术在多核平台上运行具有优秀的可扩充性，越来越多的应用软件采用了并行的设计思想。此类软件包括多媒体应用(内容创建、编辑以及本地和数据流回放)、工程和其他技术计算应用以及如应用服务器和数据库等中间层与后层服务器应用[8]。而嵌入式浏览器同样适合采用多线程的架构，因为嵌入式浏览器既有与用户交互、需要及时响应的用户界面(GUI)，又有后台复杂的加载、解析、排版、布局、渲染等操作。因此，如何对嵌入式浏览器的模块进行线程化，使其在多个处理器内核上同时运行，即设计一个多线程的嵌入式浏览器并行架构成为改进嵌入式浏览器的一个重要方向。

3　嵌入式浏览器并行机制的架构设计

在嵌入式浏览器的用户体验中，用户操作能否立即响应是用户体验的重要指标。传统浏览器中，用户在输入一个网址后，大部分时间被浏览器的后台操作所占用。这时用户要进行其他操作，都必须等浏览器的上一个网址页面显示出来才会得到响应。因此单独把浏览器的UI作为一个主线程可以给用户带来良好的用户体验。浏览器的资源下载、资源解析、排版布局、图形绘制通过定时器的异步调用来实现，因此可以把这些定时器替换为多个线程，实现一种以多线程调用代替定时器异步调用的方式进行线程划分，并且根据浏览器的后台操作模块继续划分线程。这样就会使浏览器的后台操作速度提高，在整体上提高嵌入式浏览器的性能。

根据WebKit的整体模块可知，浏览器的流程主要分为资源下载、dom树生成、render树生成、排版布局、图形渲染绘制。由于资源解析速度比较快，为了减少线程之间的开销，把资源下载和解析划分为一个线程；由于生成render树进行css匹配的时间比较耗时，把render树生成的过程与dom树生成的过程独立出来，并把render树的生成过程划分为一个线程模块，同时可以继续划分子线程处理；排版布局模块单独划分为一个线程，考虑到排版布局会涉及到很多父子节点的互相访问遍历过程，也可以在这个过程中继续划分子线程；最后图形绘制模块单独划分为绘图线程。具体嵌入式浏览器并行架构如图2所示。

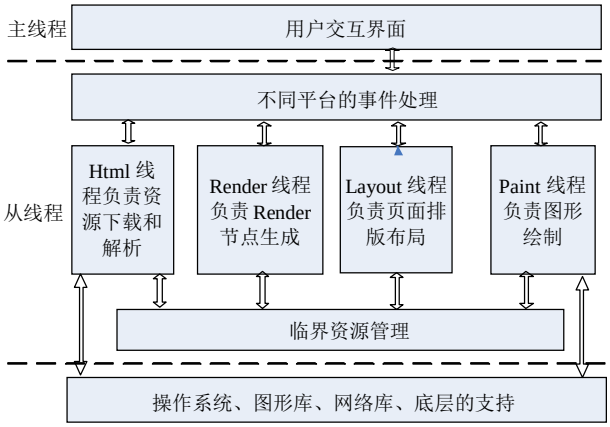

图2　嵌入式浏览器并行机制架构

3.1 线程划分背景

传统浏览器通过设置定时器异步调用各个模块完成整个浏览器的执行过程。资源下载、资源解析、排版布局、图形绘制等都进行了定时器的设置。如用户输入网址后会马上进行资源下载，此时浏览器启动资源下载定时器，WebKit生成一个定时器队列，下载定时器在定时器队列里进行排队等待，而定时时间就由系统进行定时等待。当定时时间到了，浏览器就发送一个触发资源下载的定时器事件。浏览器的定时器事件在浏览器的事件队列中进行排队，等队列中除了自己的事件都执行完成后就进行事件触发。

虽然在串行过程中，浏览器用定时器可以很好地进行各个模块的定时器异步调用，但是也占用了相当一部分事件进行定时器时间等待和定时器事件处理，以及浏览器事件的事件排队。而采用多线程方式的浏览器恰好可以去掉这些模块的定时器，等到该模块需要触发的时候，直接唤醒执行该模块的模块线程，减少了定时器的时间消耗，大大提高了嵌入式浏览器的性能。

在模块线程中也会遇到一些其他的定时器。如在资源下载时会遇到图片资源下载定时器、缓存资源管理定时器等。如果多个线程仍然进行原来的定时器处理，不仅仍然会增加定时器的等待时间，还会产生定时器资源的临界管理问题。把模块线程下面的这些定时器所执行的事件放在模块线程下面的事件队列中，当要触发这些事件时，直接去模块线程中增加这些事件即可，从而可以合理地去掉这些模块线程下的定时器。

3.2 资源下载和解析

用户输入网址后，浏览器的下载模块进行资源下载，下载一段资源后马上交给浏览器进行解析生成dom树。如果用户所请求的网址资源比较大，下载占用的时间会变长。下载的过程可以单独划分为一个线程执行，又由于下载后资源会马上解析，解析的速度很快，并且解析模块与资源下载模块的耦合度比较高，因此把解析模块和下载模块放在一个线程来处理。这样用户在输入网址后，浏览器马上交给下载线程来处理资源下载和资源解析的过程，等解析出一段dom树后，浏览器又可以继续进行线程交互。

3.3 Render树生成

WebKit把下载的资源解析为dom节点[9]时会立刻进行css匹配生成render节点[10]，这个过程需要一定时间，此时，浏览器会根据当前节点的号码去css信息表中根据css类型进行查找该节点所对应的具体css信息进行匹配。这样每个节点必须生成render节点后，才会解析下一个节点。为了减少这一模块的时间，把dom生成和render生成的过程分离出来，即浏览器生成dom节点后不立即生成render节点，而是继续生成下一个dom节点。在浏览器布局之前，或者需要进行render树生成时，把dom节点批量生成render节点，因此把render树的生成过程划分为一个线程。

每一个dom节点都要生成render节点，通过主分线程来处理，并可以在这个模块继续划分子线程。render线程负责该render树生成模块，进行总体管理。第一个子线程会对第一个dom节点进行render生成，第二个子线程会对第二个dom节点进行render生成，当子线程生成当前节点为render节点后通知render线程，render线程为其分配下一个需要进行css匹配生成的dom节点。

3.4 布局计算

网页排版技术负责合理安排网页中每个控件的显示位置、及时动态地响应网页内容的变化并对网页显示控件重新布局排版，是衡量浏览器显示效果及速度的一个重要指标[11]。浏览器进行布局计算过程可以划分为一个线程来完成，并且可以继续划分子线程。当父节点访问子节点的位置信息时对子节点进行遍历，对每个子节点的遍历是相互独立的过程，因此可以划分多个子线程；当子节点排版布局时会访问父节点的位置信息，同一父节点的子节点的访问过程也是相互独立的，也可以划分多个子线程。

3.5 图形绘制

当网页信息排版布局后浏览器调用绘图模块进行绘制，从而呈现出整个页面。绘图任务可以单独划分为一个从线程来完成。当浏览器需要绘图时就向绘图线程发送一个事件唤醒绘图线程，绘图线程检测到线程事件队列中的绘图事件就发出绘图操作。

3.6 临界资源管理

在多线程浏览器设计中，会碰到很多临界资源处理的问题。这些临界资源主要是dom数据和render数据，还有其他的显示框架、绘制区域、css样式等。需要对临界的资源进行加锁控制，保证其在多核线程之间的正确处理，具体如图3所示。

Html线程下载解析完后会生成dom节点，这些dom节点所对应的render节点可能被排版布局线程所使用进行排版布局，或者被render线程使用进行

dom节点样式匹配，或者被图形绘制线程使用进行网页渲染及字体样式加载[12]。此时如果因为其他事件导致该dom节点发生变化，那么当前的这个dom节点会被释放，对应的render节点也会被释放，而由新的dom节点或render节点代替，如果不对该dom数据、render数据进行临界资源管理，其他正在使用该render节点的的线程则会发生访问失败。

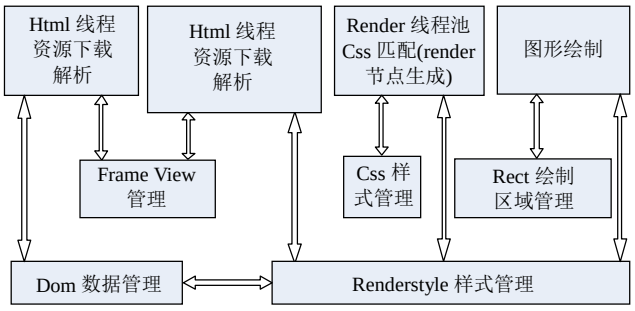

图3　临界资源管理架构图

此外，除了dom数据和render数据外还有其他临界资源。Html线程进行资源下载后会对frameview模块[13]进行资源更新，当前的frameview模块会被马上释放，而同时用这个frameview模块的还有排版布局线程。图形绘制时，必须要保证同一个时刻只有一个线程在操作图形绘制区域，否则在把完成绘图的图形绘制区域清除的同时可能还会把没有绘图的区域也一同清除掉。

4　实验验证

4.1　具体实现

线程划分方式以上述方式划分，采用pthread函数线程库实现具体线程调用，以mutex加锁以及条件变量、事件发送实现数据同步。

4.2　实验环境

硬件环境：某公司多核数字电视机顶盒(处理器主频500 MHz)，数字电视及相应配件。

软件环境：Linux 平台，skia图形库，sdl事件支持和基于WebKit的全功能嵌入式浏览器。

4.3　功能测试

表1　浏览器技术支持标准

技术标准	版本	支持情况
Html	4.0	支持
Html	5.0	支持
Javascript	V1.4	支持
图片	Gif	支持
图片	Png	支持
图片	Jpeg	支持
插件	Flash	不支持
http	V1.1	支持

通过打开网页，对嵌入式浏览器多线程结构、网页浏览等基本功能测试显示效果如图2所示。通过测试，嵌入式浏览器对相关技术标准支持的具体情况如表1所示。

4.4　性能测试

浏览器的数据测试采用对串行浏览器和多核架构的并行浏览器打开10个网页求平均值进行速度测试和占用内存测试，从而能避免网速等其他条件的偶然影响。

测试用户从输入网址到网页内容显示的时间数据如表2所示，测试用户从输入网址到网页显示的内存消耗比较如表3所示。对测试数据进行分析可得知，与现有串行浏览器相比，多线程架构的嵌入式浏览器打开网页的速度提升为18%～40%，而由于多个线程的存在，对系统资源的占用也随之有所增加。其中，CPU占用率的增加会由于多核平台的逐渐普及得到解决，而内存的消耗属于将多线程引入浏览器架构的必然负面结果，但同样在可接受范围内。

表2　打开同一网页的浏览器时间比较

网页	原始版本/MB	多线程版本/MB	内存增加百分比/(%)
百度	1.13	0.95	18.94
新浪	5.58	3.81	32.00
搜狐	4.29	3.35	28.05
腾讯	5.50	3.95	39.24

表3　打开同一网页的浏览器内存消耗比较

网页	原始版本/MB	多线程版本/MB	内存增加百分比/(%)
百度	23.41	25.12	7.30
新浪	52.84	57.38	8.59
搜狐	43.02	46.85	8.90
腾讯	49.29	53.39	8.32

5　结　论

利用本文方法，嵌入式浏览器在多核处理器上以一定的内存运行范围内，对网页显示有较快的速度。同时在嵌入式浏览器的资源加载、解析、排版布局、图形绘制方面也有一定的理论参考价值。经试验验证，嵌入式浏览器采用该并行方法支持大多数技术标准。在提高网页显示速度的同时，减小嵌入式浏览器的内存，也是完善该方法进一步要做的工作。

参　考　文　献

[1] 赵经纬, 周余, 王自强, 等. 基于WebKit的嵌入式浏览器的研究与实现[D]. 南京: 南京大学, 2009.
ZHAO Jing-wei, ZHOU Yu, WANG Zi-qiang, et al. Research and implementation of embedded browser based on WebKit[D]. Nanjing: Nanjing University, 2009.

[2]邢文华. 基于WAP的嵌入式浏览器的关键技术的研究[D]. 天津: 河北工业大学, 2011.

XING Wen-hua. The key technology research of embedded browser based on Wap[D]. Tianjin: Hebei University of Technology, 2011.

[3] DALI L, RUSU D, MLADENI D. Enhanced Web page content visualization with Firefox[D]. Austria: Jožef Stefan Institute, 2009.

[4] Google. The Chromium projects[EB/OL]. [2012-12-10]. http://ww w.chromium.org.

[5] WetKit. The WebKit open source project[EB/OL]. [2011-02-10]. http://www.webkit.org.

[6] 史君, 桑国明. 基于嵌入式移动浏览器的QtWebkit 2.1内核研究与开发[D]. 大连: 大连海事大学, 2012.
SHI Jun, SANG Guo-ming. Research and development of QtWebkit 2.1 kernel based on embedded mobile browser. [D]. Dalian: Dalian Maritime University, 2012.

[7] 王映, 于满泉, 李盛韬, 等. .JavaScript引擎在动态网页采集技术中的应用[D]. 北京: 中国科学院计算技术研究所, 2004.
WANG Ying, YU Man-quan, LI Sheng-tao, et al. Extracting dynamic URLs using JavaScript engine[D]. Beijing: Software Lab, Institute of Computing Technology, Chinese Academy of Science, 2004.

[8] GALENSON J, JONES C G., LO J. Towards a browser OS[D]. San Francisco: University of California Berkeley, 2008.

[9] DUBEY S. AJAX performance measurement methodology for internet explorer 8 Beta 2[J]. CODE Magazine, 2008, 5(3): 53-55.

[10] JONES C, LIU R, MEYEROVICH L, et al. Parallelizing the Web browser[D]. San Francisco: University of California Berkeley, 2009.

[11]刘敏. 嵌入式浏览器网页排版技术研究与实现[D]. 武汉: 华中科技大学, 2011.
LIU Min. A thesis submitted in partial fulfillment of the requirements for the degree of master of engineering[D]. Wuhan: Huazhong University of Science and Technology, 2011.

[12] TURNER D. The design of FreeType2[EB/OL]. [2012-12-10]. http://www.freetype.org.

[13] MEYEROVICH L, BODIK R. Fast and parallel Webpage layout[D]. San Francisco: University of California Berkeley, 2009.

编 辑 漆 蓉