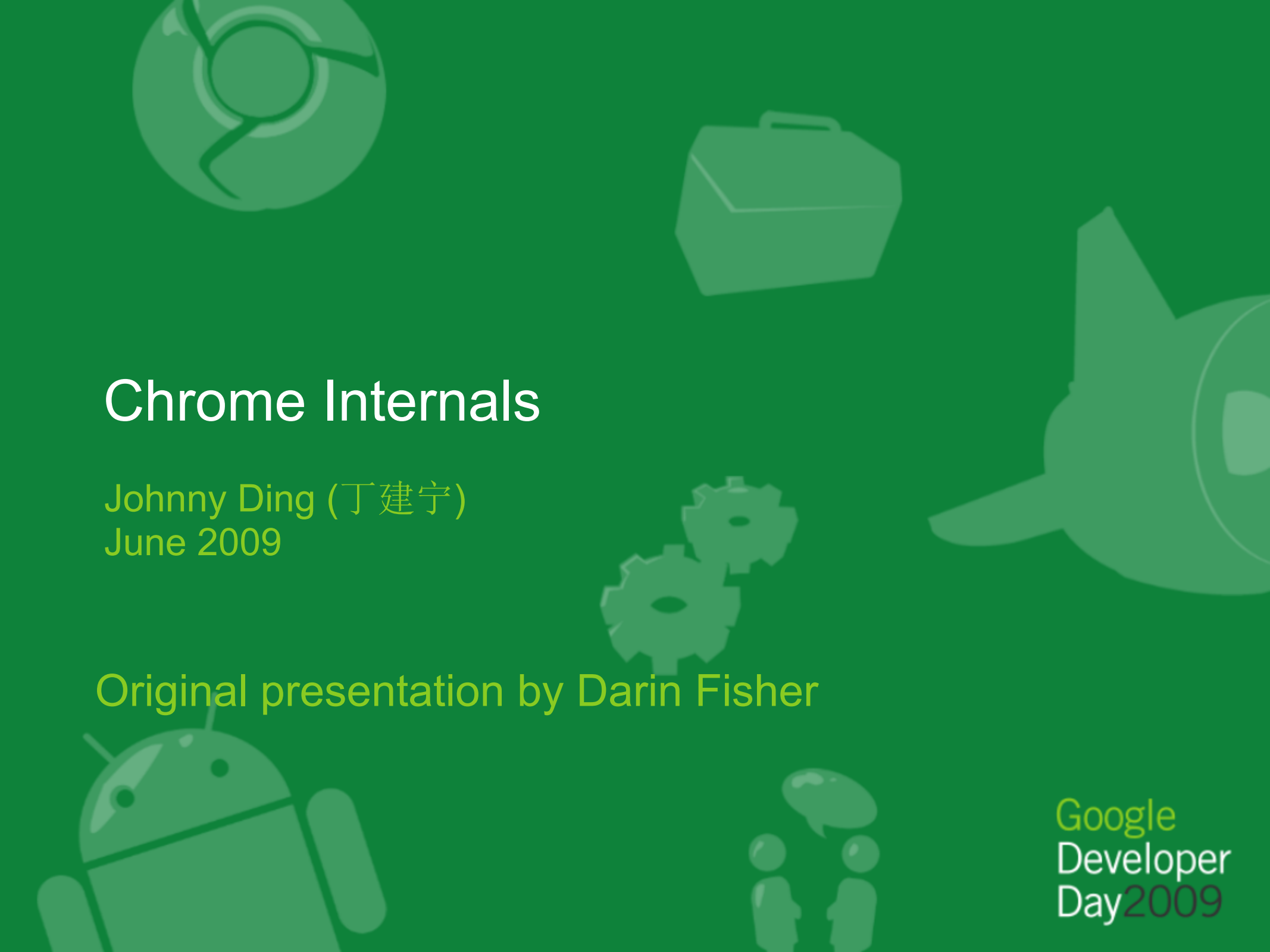

Google
Developer
Day2009



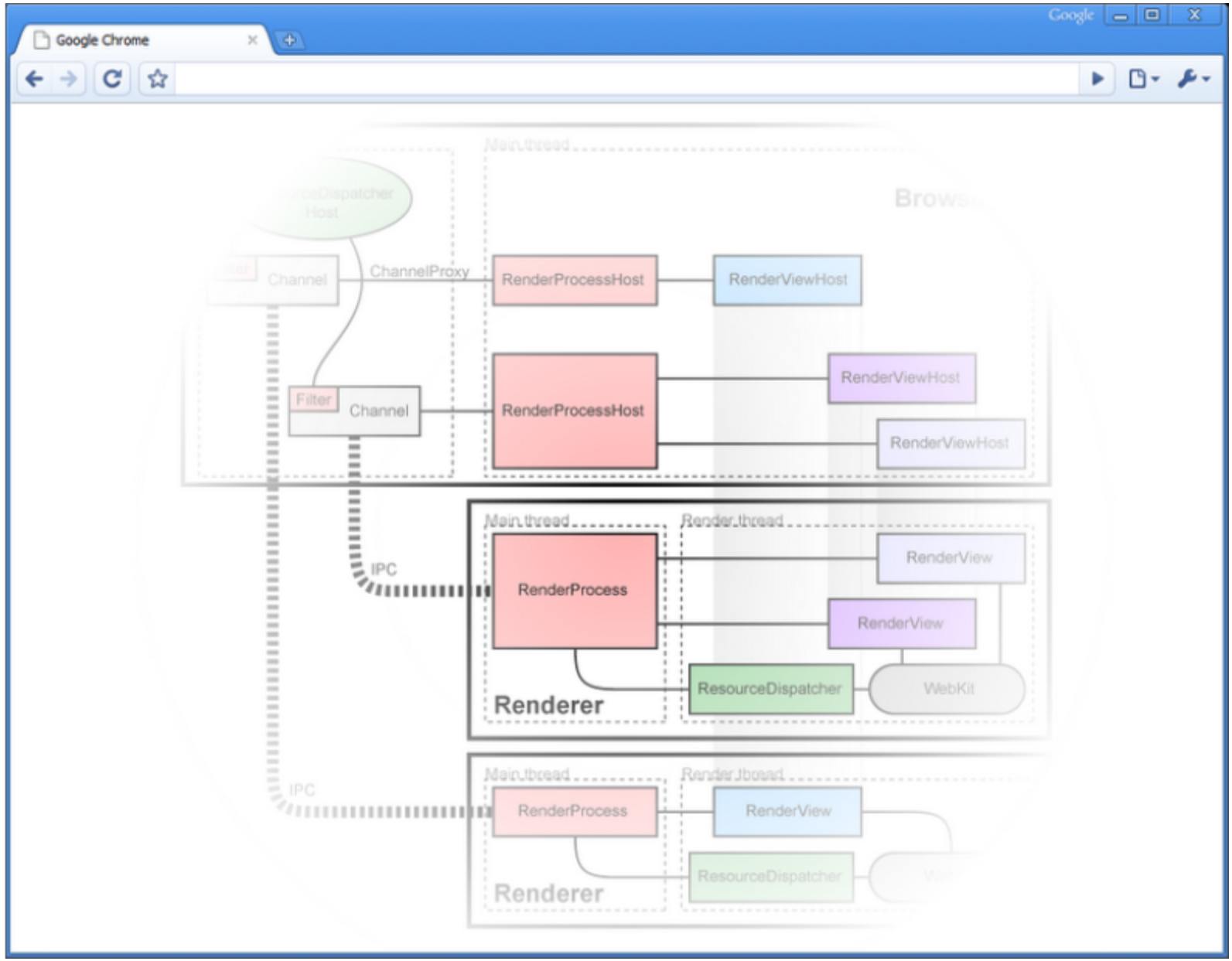
Chrome Internals

Johnny Ding (丁建宁)
June 2009

Original presentation by Darin Fisher

Google Developer Day 2009


简洁界面，强力核心
Google
Developer
Day2009


Google Chrome
Main thread
Channel
ChannelProxy
RenderProcessHost
RenderViewHost
Filter
Channel
RenderProcessHost
RenderViewHost
RenderViewHost
Main thread
Render thread
IPC
RenderProcess
RenderView
RenderView
ResourceDispatcher
WebKit
Renderer
IPC
RenderProcess
RenderView
ResourceDispatcher
Renderer

“现代浏览器不仅仅是浏览器，它更像一个协同式多任务操作系统.”

设计指导原则, 2006

目标

- 速度
- 稳定
- 安全

使用多进程架构!

- 速度: 不同的网络应用分别运行在独立的线程
- 稳定: 不同的网络应用分别运行在各自独立的地址空间
- 安全: 使用Sandbox技术为网络应用提供隔离的运行环境

速度是王道

- WebKit
  - 超快，开源的布局引擎 (Rendering Engine)
  - 较小的资源占用 (在嵌入式浏览器上的广泛使用)

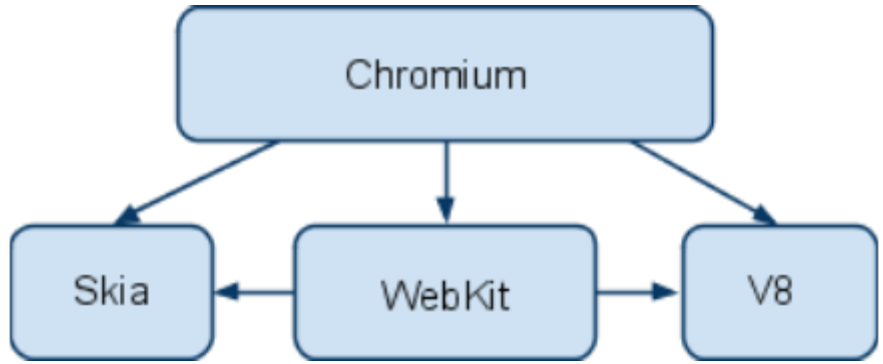
- V8
  - 超快优化的V8 JavaScript 引擎
  - 网络应用可以以更快的速度运行复杂的逻辑

内核剖析...

Google
Developer
Day2009

主要模块

- Chromium
  - UI: tab 控件, 多功能地址栏, new tab page, ...
  - 多进程架构
  - 历史数据管理
  - 网络栈
  - Sandbox
  - Safe Browsing
  - …
- Skia
- WebKit
- V8

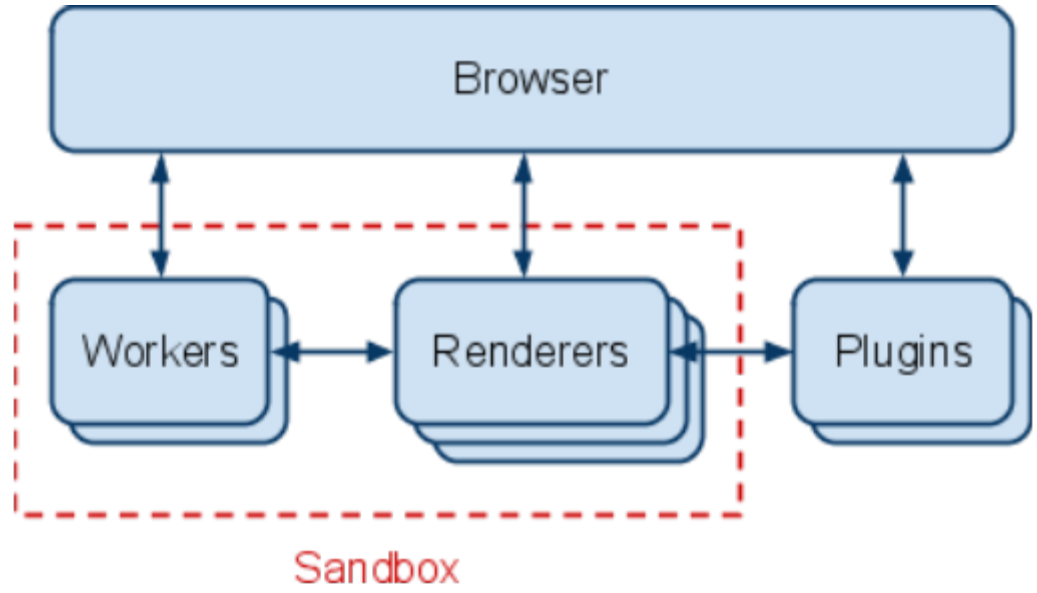


多进程架构

Google
Developer
Day2009

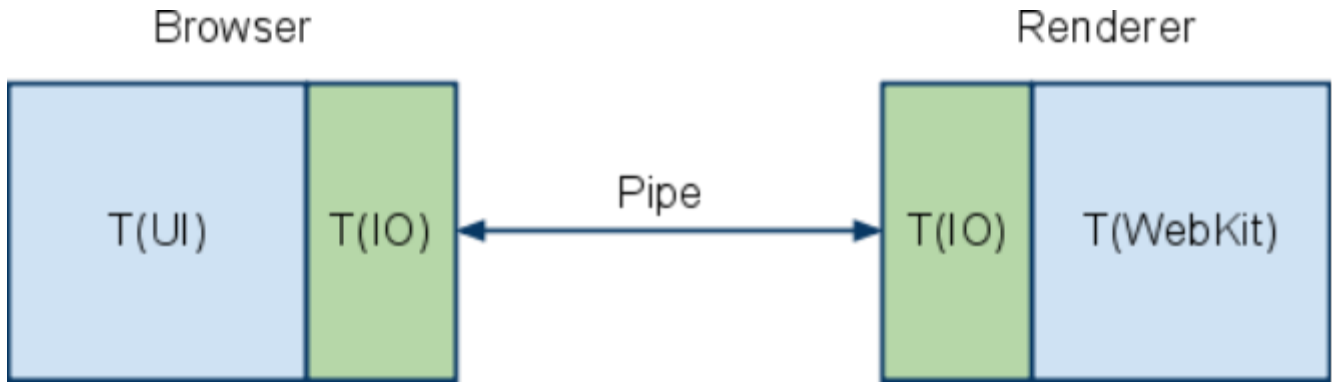
进程类型

- 浏览器
  - 主控和协调各功能模块和相关进程
  - IO 处理和转发
  - Trusted
- Renderer, Worker
  - 嵌入WebKit来解析, 渲染和处理网页及网络应用
  - **Untrusted**
- 插件:
  - 嵌入NPAPI插件 (Flash, Java, Silverlight, etc.)
  - Trusted :-(



进程间通信

- 通信方式
  - 使用 named pipes来作为IPC通信通道，以异步通信方式为主。
  - 少量的IPC通信使用同步方式。
- 有些数据使用Shared Memory(共享内存)来相互交换
- 每个进程有一个专门的线程来处理IPC通信：



Tab的进程模型

- 每个Tab使用一个进程
- 多Tab共享一个进程:
  - 多Tab间存在潜在的脚本交互关系
  - 点击链接时打开新Tab，例如<a target=“_blank” href=“…” >…
  - 达到最大使用进程数限制
- 在地址栏上敲入一个与当前Tab不同域名的地址, 一个新的进程会产生并赋予该Tab. 旧的进程将会被终止。
- 多个同属一个域名的Tab共享一个进程, 但是...

Sandbox 技术

- 主要目标:
  - 保护用户的系统不被网页上的恶意软件侵入
- 目标程序将受到限制:
  - 控制其对文件系统和网络的访问
  - 限制其对窗口系统的访问
  - 限制其对输入设备的访问
- 使用方法:
  - 基于user token产生的受限token
  - 基于受限job对象的进程
  - 将目标程序运行在一个虚拟桌面内

Sandbox 技术

- 但是，浏览器有时的确需要访问文件系统!
  - 需要支持文件上传
  - 需要支持访问文件协议file://类URL

- Sandbox不能保护
  - Cookies
  - Passwords
  - HTML5 数据库, local/session storage
  - 跨站攻击 (XSS, 用户数据存贮在网络上)

在sandbox中渲染页面

- 主要流程:
  - 渲染进程渲染页面到位图
  - 发送该位图到浏览器进程
  - 浏览器拷贝该位图到屏幕

- 难点:
  - 对某些操作系统API的访问收到限制 (fonts, etc.)
  - 大量的渲染行为不应该锁住浏览器进程
  - 更快!更快!更快!

更新页面显示与滚动页面显示

- 锁无关的页面更新(lock free):
  - 浏览器进程为页面维护一个渲染位图的后备存储
  - 渲染进程通过Shared Memory(共享内存)发送页面渲染的更新位图到后备存储
  - 浏览器进程从后备存储读取页面位图并画在屏幕上
  - 浏览器进程返回ACK响应到渲染进程，通知渲染进程可以发送新的渲染信息更新

- 滚动页面显示使用类似方法(传送滚动区域的位图)

资源装载

- 其他进程(插件进程，渲染进程)使用浏览器进程作为代理请求相关的IO操作(资源请求)。浏览器进程
  - 提供对 file:// 和 chrome:// 协议URL的访问
  - 对网络资源请求进行 safe-browsing处理
  - 处理cookies

- 在WebKit真正得到数据之前，浏览器进程...
  - 处理HTTP重定向
  - 处理HTTP认证
  - 探测MIME类型 (为了在浏览器进程端的下载管理)
  - 为SSL链接提供安全检查

历史信息管理

- 锁无关的 visited links系统
  - 通过共享内存让其他进程可以读取visited links
  - 使用URL Hash作为索引
  - 仅仅浏览器进程可以更新visited links
  - 通过建立一个新的copy来调整visited links的大小

- 当页面装载完毕,
  - 文本被从页面抽取用于建立全文搜索(Full Text Search)的索引(使用sqlite)
  - 产生页面缩略图并保存

插件

- 支持的类型:
  - Netscape style plugins
  - Whitelist of ActiveX controls (only WMP now)
- 相同插件共享一个进程
  - 单进程浏览器使用相同的模型
  - 有些插件需要较长的时间来装载
- 挑战: NPAPI是同步调用的API
  - 缓存windowless插件的rendering
  - 对windowed类型插件，做了很多的优化
  - 移植到不同平台!

WebKit





Google
Developer
Day2009

WebKit概况

- 主要模块:
  - JavaScriptCore: JS engine (未使用)
  - WebCore: HTML+CSS rendering, DOM, etc.
  - WebKit: embedding API layer (未使用)

- WebCore的Chromium代码:
  - PLATFORM(CHROMIUM) ⇨platform/chromium
  - PLATFORM(SKIA) ⇨platform/graphics/skia
  - USE(V8) ⇨bindings/v8

- WebKit版本:
  - Chrome 1 ~ Safari 3
  - Chrome 2 ~ Safari 4

WebKit 开发

- 基于WebKit的Chromium开发
  - 三个reviewers
  - 十多个contributors
- 现况: **Unforked!!**
- 工作重点:
  - 基于Chromium的WebKit API
  - 开放式网络平台 (HTML5, etc.)
  - 改善网站兼容性问题
  - 性能

开放式网络平台 HTML5

- 正在开发的一些功能:
  - Audio/video
  - Application caches
  - Database
  - Local storage
  - Session storage
  - Notifications
  - Web workers: dedicated, persistent, shared
- 多进程架构和Sandbox技术为浏览器开发带来挑战

网络栈实现





Google
Developer
Day2009

为了更快更好

- 从 Wininet 到 Winhttp 到 src/net/http/
- DNS 预读取/缓存
- 正在开发:
  - Feature parity (客户端认证, socks, IPv6 literals(RFC2732))
  - Sparse caching
  - Pseudo-pipelining
  - Deferred connection binding
  - Parallel proxy auto config


Learn more at code.google.com
Google
Developer
Day2009


Q&A
Google
Developer
Day2009