



























则会根据别名资源来构造响应信息返回给浏览器;否则,资源映射器会发送网络请求,交由原 Web 服务器处理(图 4 中的⑧),当响应信息构造完成后,终端代理服务器随即将其返回给浏览器(图 4 中的⑨)。

本文的原型系统目前只针对采用 HTTP 协议的 Web 应用.由于 HTTPS 的默认机制不支持代理、个人云等中间节点,因此目前实现无法直接支持采用 HTTPS 协议的 Web 应用.一种解决方案是将代理服务器设置为可信中间节点,即浏览器和终端代理服务器以及云端代理服务器和原服务器之间可以采用两个独立的 HTTPS 连接进行通信;对于旁路机制,终端代理服务器可以通过一个新的独立 HTTPS 连接请求原服务器。

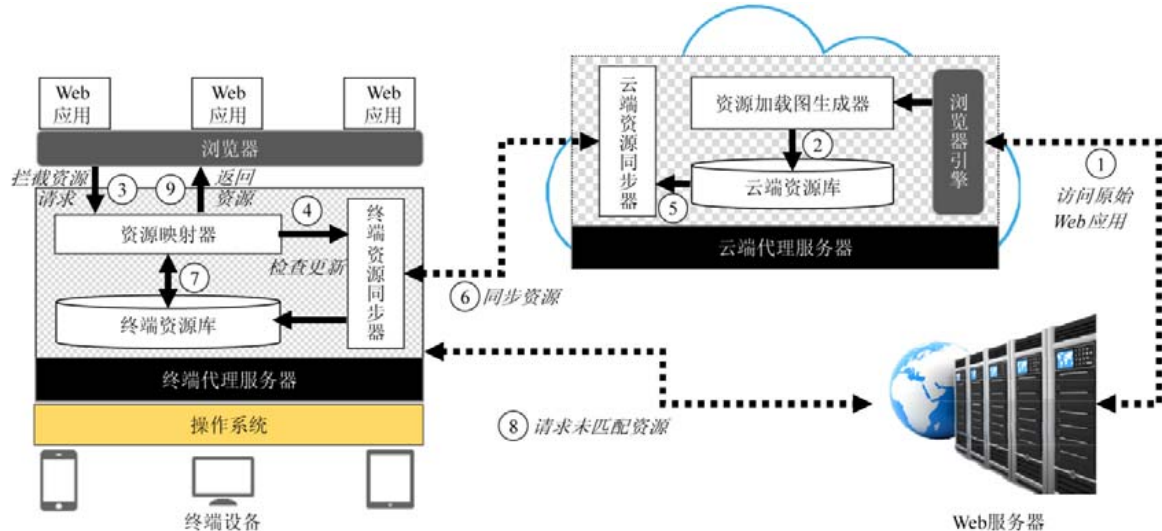


Fig.4 The architecture of SWAROVsky system

图 4 SWAROVsky 系统架构

## 4 效果评估

本文从 Alexa 排名前 500 的移动 Web 应用中选取了 50 个应用的首页作为开展实验评估的对象,采用仿真实验的方法来精确评估缓存性能的提升.数据收集方法与第 1 节的度量实验类似,通过以较短的周期收集移动 Web 应用在一段时间的资源演化情况,可以仿真用户以不同频率访问移动 Web 应用的行为,进而通过理论计算或实际测量的方式比较原有机制和本文实现的 ReWAP 和 SWAROVsky 系统的缓存性能.本文采用节省的数据流量来定量评估缓存性能的提升。

### 4.1 ReWAP 系统性能评估

由于 ReWAP 需要对现有移动 Web 应用的实现进行少量修改,在没有应用源码的情况下,本文采用仿真计算的方法来分析 ReWAP 所带来的缓存性能提升.通过将收集到的移动 Web 应用资源演化序列,可以计算出在每一时刻的资源包配置文件.接着,通过维护一个理想的浏览器运行仿真器和一个 ReWAP 运行容器的仿真器,可以为每个 Web 应用计算用户在不同再访问间隔下访问时的具体流量消耗,从而可以比较二者之间的流量差异.本文以 30 分钟为粒度、评估用户在再访问间隔从 30 分钟到 1 天的各种情况下(即,30 分钟、1 小时、1.5 小时、...、24 小时)ReWAP 为用户节省流量的平均值。

图 5 展示了实验结果.在 0.5 小时~24 小时的再访问间隔下,相比于传统的浏览器缓存机制,ReWAP 在平均情况下能够节省 8%~51%之间的流量.随着再访问间隔的增加,节省的流量逐渐变少,其原因是资源包发生了变化,需要通过网络更新,带来额外的流量消耗.对于每个再访问间隔,流量节省的分布方差都非常大.在最优情况下,尤其是再访问间隔在 5 小时以内时,流量节省的最大比例甚至接近 100%.相反地,当再访问间隔在 14 小时以上时(图中编号为 29~49 的箱线图),最大的节省流量比例仅为 50%。

为了评估 ReWAP 系统的额外性能开销,本文采用生成测试页面的方式来定量测量.这些测试页面包含的资源个数从 20~100 不等,每个资源大小都是 100KB,采用 JavaScript 的方式动态获取.假设页面的所有资源都配置到资源包中,并且有 10%的资源是通过归一化得到的别名资源.我们为每个测试页面生成对应的 ReWAP 版本,然后依次访问这些测试页面.在每个页面的访问过程中,通过 Android Monitor 工具记录进程的 CPU 和内存负载.结果表明,相比于原始页面,采用 ReWAP 后的应用 CPU 负载增加了 15%,内存没有显著变化.

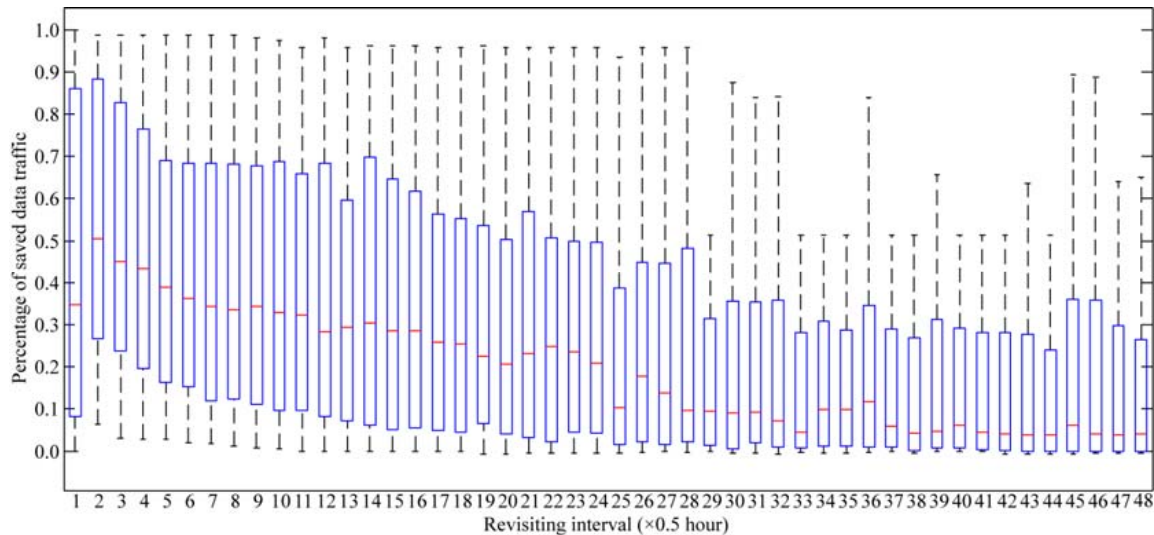


Fig.5 The distribution of saved network traffic under different revisiting intervals by ReWAP

图 5 不同再访问间隔下 ReWAP 节省网络流量的分布

#### 4.2 SWAROVsky 系统性能评估

SWAROVsky 系统基于双代理架构实现基于云-端融合的缓存优化,无需对 Web 应用进行修改,所以可在真实环境下评估效果.本文在局域网环境部署了 SWAROVsky 系统,终端使用三星 N7100 手机,采用 Firefox 浏览器来访问移动 Web 应用.云端代理服务器部署在一台联想 T420i 笔记本电脑上.电脑与手机连在同一个 WiFi 下.本文通过网络控制软件为手机和电脑之间设置了 150ms 的网络延迟,以仿真手机的真实网络环境.为了验证在不同再访问间隔的情况下 SWAROVsky 系统的有效性,本文将 50 个 Web 应用的资源数据部署在一台回放服务器上,作为 Web 应用的服务器为前端访问提供资源.

针对第 1 次访问 Web 应用和以不同再访问间隔再次访问同一 Web 应用两种情况,我们测量了使用浏览器直接访问和采用 SWAROVsky 系统访问时的网络流量,进而可以计算流量减少的比例.图 6 展示了实验结果,横坐标代表流量减少比例,纵坐标代表累积分布.采用 SWAROVsky 系统后,Web 应用在第 1 次访问时网络流量减少比例的中位数是 4%,原因是系统的数据流压缩可以减少网络流量.在大约 20%的情况下网络流量有略微增加,其原因是系统在资源同步时需要传输额外信息.当 Web 应用被再次访问时,网络流量减少比例的中位数是 57.6%.不同再访问间隔下的减少比例有所差异,整体来看,SWAROVsky 系统对较短再访问间隔下的性能提升高于较长再访问间隔.

在开展仿真实验的同时,我们记录了终端代理服务器和云端代理服务器的 CPU 和内存使用情况,可以用来评估 SWAROVsky 系统的运行开销.终端代理服务器的 CPU 使用率平均为 2.7%,仅仅是 Firefox 的 1/10.而且在实验过程中,终端代理服务器的 CPU 使用率并没有发生明显变化,而 Firefox 的 CPU 使用率变化较为显著.终端代理服务器的内存使用率同样仅为 Firefox 浏览器的 1/10.随着 Web 应用访问的增加,终端代理服务器的内存使用率变化也不是特别明显.这些结果表明了终端代理服务器是轻量级的,不会对浏览器造成额外的用户体验损耗.对于云端代理服务器,资源同步器的初始内存使用是 37MB,同步资源时的内存使用大约为 70MB.资源加载

图生成器的内存使用为 100MB,当访问 100 个 Web 应用之后,内存使用增加约 30%,CPU 的使用率为 10%.总体来讲,云端代理服务器的运行开销较小,适合在个人云上部署.

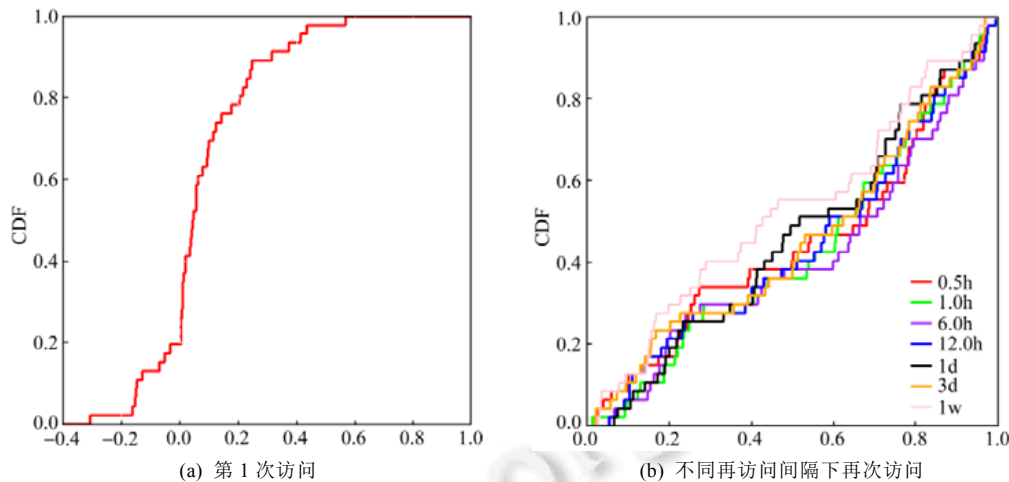


Fig.6 The distribution of saved network traffic by SWAROVsky

图 6 SWAROVsky 节省网络流量的分布

## 5 相关工作

缓存是优化 Web 应用性能的重要手段.在 PC 时代,Web 缓存主要关注服务器端和网络缓存,以减少整个互联网的网络负载,主要技术手段包括代理缓存(proxy caching)、反向代理缓存(reverse proxy caching)、多级缓存(caching hierarchy)等,研究工作主要集中在缓存服务器结构、内容替换策略和缓存一致性问题<sup>[10,11]</sup>.

随着移动互联网的兴起,移动计算设备已经成为互联网访问的第一入口.移动计算设备经常通过电信蜂窝网络接入互联网.蜂窝网络带宽小、延迟高、不稳定、按流量计费的特性,使得浏览器缓存的重要性日益突出.研究者对移动 Web 应用的浏览器缓存性能开展了度量实验.Wang 等人<sup>[12]</sup>根据 24 个 iPhone 手机用户 1 年的浏览器访问数据分析了浏览器缓存性能,发现 60% 的 Web 资源无法在浏览器缓存中命中.Qian 等人<sup>[13]</sup>根据基站收集的 1 天 Web 流量数据和 20 个手机用户 5 个月的 Web 浏览记录,研究了移动浏览器缓存的实现缺陷所导致的冗余传输问题,他们发现,两个数据集中分别有 18% 和 20% 的 Web 流量是冗余的,进而分析了缓存大小限制、未完全遵循 RFC 2616 规范等 12 个原因.Qian 等人的后续工作<sup>[14]</sup>揭示了 500 个流行网站的缓存低效问题:26% 的根 HTML 文件无法缓存、23% 的资源被配置了小于 1 小时的过期时间.Wang 等人<sup>[15]</sup>还研究了以 HTML、CSS、JS 资源的具体内容为判断依据的细粒度缓存的性能,发现在 Web 资源较长时间内仅有部分内容发生变化,验证了细粒度缓存的有效性.

针对移动 Web 应用的浏览器缓存性能问题,Wang 等人<sup>[12]</sup>提出推测式加载机制来支持同一 Web 应用的不同页面的缓存复用.Zhang 等人<sup>[16]</sup>在移动操作系统层设置专门的缓存管理模块来处理 HTTP 请求,解决浏览器缓存在实现方面的问题.Lyberopoulos 等人<sup>[17]</sup>和 Butkiewicz 等人<sup>[18]</sup>根据用户的 Web 浏览行为建立预测模型,对用户将要访问的 Web 应用通过预取(prefetching)的方式提前刷新缓存,从而加快应用加载时间.然而,预取的内容如果没有被用户访问则会造成额外的流量开销.针对此问题,Baumann 等人<sup>[19]</sup>设计了选择性预取策略来决定预取的时机和内容.Pande 等人<sup>[20]</sup>利用 HTML 5 的 Service Worker 机制实现对移动 Web 应用缓存资源的精确更新.Netravali 等人<sup>[21]</sup>提出远程控制缓存思想,通过在服务器端重写 URL 以解决别名资源问题.

对比相关工作,本文的度量实验没有依赖被动收集的用户 Web 浏览数据集,而是采用主动式方法,研究 Web 应用的资源演化和缓存配置对浏览器缓存性能造成的影响,进而优化用户频繁周期性访问相同 Web 应用时的缓存性能.在缓存性能优化的总体效果上,本文方法节省的网络流量与近期相关工作<sup>[21]</sup>的水平相当.

## 6 总结与展望

本文设计并开展了主动式、规模化的移动 Web 应用浏览器缓存性能度量实验,揭示了浏览器缓存的理论性能上限和实际性能之间的巨大差距,发现了重复请求别名资源、启发式过期时间和保守的过期时间配置这三大影响浏览器缓存性能的原因.针对发现的问题,提出了两种浏览器缓存性能优化方法并实现了原型系统:一种是在应用层基于资源包的缓存优化方法,将 Web 应用相对稳定的资源配置到资源包中,利用 HTML 5 的应用缓存机制实现从资源包加载资源;另一种是在平台层基于云-端融合的缓存优化方法,通过云端预先加载 Web 应用获取资源状态,然后与终端的缓存资源进行同步,实现缓存的精确控制.实验结果表明,用户在 1 天内再次访问相同 Web 应用时,采用本文提出的两种方法分别平均可减少 8%~51%和 4%~58%的数据流量,且系统开销较小.

基于本文工作,未来可以通过分析和预测用户访问的 Web 应用,进一步优化浏览器缓存的整体性能.同时,可以考虑细粒度缓存方法,通过分解 Web 资源的具体内容,实现增量更新,减少缓存失效时的更新开销.

### References:

- [1] ComScore. Mobile Internet usage Skyrockets in past 4 years to overtake desktop as most used digital platform. <https://www.comscore.com/Insights/Blog/Mobile-Internet-Usage-Skyrockets-in-Past-4-Years-to-Overtake-Desktop-as-Most-Used-Digital-Platform>
- [2] Huang G, Liu X, Zhang Y. A mobile Web application platform with synergy of cloud and client. *SCIENCE CHINA Information Sciences*, 2013,(1):24–44 (in Chinese with English abstract).
- [3] Serrano N, Hernantes J, Gallardo G. Mobile Web apps. *IEEE Software*, 2013,30(5):22–27.
- [4] Wang X, Balasubramanian A, Krishnamurthy A, Wetherall D. Demystifying page load performance with Wprof. In: *Proc. of the 10th USENIX Conf. on Networked Systems Design and Implementation (NSDI 2013)*. 2013. 473–486.
- [5] Li D, Hao S, Gui J, William H. An empirical study of the energy consumption of Android applications. In: *Proc. of the 30th IEEE Int'l Conf. on Software Maintenance and Evolution (ICSME 2014)*. 2014. 121–130.
- [6] Alexa. <http://www.alexa.com/>
- [7] RFC 2616. <http://www.w3.org/Protocols/rfc2616/rfc2616.txt>
- [8] Zhang T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: *Proc. of the 21st Int'l Conf. on Machine Learning (ICML 2004)*. 2004. 919–926.
- [9] Application cache. <https://www.w3.org/TR/html5/browsers.html#offline>
- [10] Wang J. A survey of Web caching schemes for the Internet. *ACM SIGCOMM Computer Communication Review*, 1999,29(5):36–46.
- [11] Barish G, Obraczke K. World Wide Web caching: Trends and techniques. *IEEE Communications Magazine*, 2000,38(5):178–184.
- [12] Wang Z, Lin X, Zhong L, Chishtie M. How far can client-only solutions go for mobile browser speed? In: *Proc. of the 21st Int'l Conf. on World Wide Web (WWW 2012)*. 2012. 31–40.
- [13] Qian F, Quah KS, Huang J, Erman J, Gerber A, Mao Z, Sen S, Spatscheck O. Web caching on smartphones: Ideal vs. reality. In: *Proc. of the 10th Int'l Conf. on Mobile Systems, Applications, and Services (MobiSys 2012)*. 2012. 127–140.
- [14] Qian F, Sen S, Spatscheck O. Characterizing resource usage for mobile Web browsing. In: *Proc. of the 12th Int'l Conf. on Mobile Systems, Applications, and Services (MobiSys 2014)*. 2014. 218–231.
- [15] Wang X, Krishnamurthy A, Wetherall D. How much can we micro-cache Web pages? In: *Proc. of the 2014 Conf. on Internet Measurement (IMC 2014)*. 2014. 249–256.
- [16] Zhang Y, Tan C, Qun L. CacheKeeper: A system-wide Web caching service for smartphones. In: *Proc. of the 2013 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing (UbiComp 2013)*. 2013. 265–274.
- [17] Lymberopoulos D, Riva O, Strauss K, Mittal A, Ntoulas A. PocketWeb: Instant Web browsing for mobile devices. In: *Proc. of the 17th Int'l Conf. on Architectural Support for Programming Languages and Operating System (ASPLOS 2012)*. 2012. 1–12.
- [18] Butkiewicz M, Wang D, Wu Z, Madhyastha HV, Sekar V. KLOTSKI: Reprioritizing Web content to improve user experience on mobile devices. In: *Proc. of the 12th USENIX Symp. on Networked Systems Design and Implementation (NSDI 2015)*. 2015. 439–453.



- [19] Baumann P, Santini S. Every byte counts: Selective prefetching for mobile applications. In: Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT). 2017,1(2):6:1-6:29.
- [20] Pande N, Somani A, Samal SP, Kakkirala V. Enhanced Web application and browsing performance through service-worker infusion framework. In: Proc. of the 2018 IEEE Int'l Conf. on Web Services (ICWS 2018). 2018. 195-202.
- [21] Netravali R, Mickens J. Remote-control caching: Proxy-based URL rewriting to decrease mobile browsing bandwidth. In: Proc. of the 19th Int'l Workshop on Mobile Computing Systems & Applications (HotMobile 2018). 2018. 63-68.

附中文参考文献:

- [2] 黄罡,刘讚哲,张颖.面向云-端融合的移动互联网应用运行平台.中国科学:信息科学,2013,(1):24-44.



马郢(1989-),男,博士,助理研究员,CCF 专业会员,主要研究领域为系统软件,Web 系统,移动计算.



梅宏(1963-),男,博士,教授,博士生导师, CCF 会士,主要研究领域为系统软件,软件工程.



刘讚哲(1980-),男,博士,副教授,博士生导师,CCF 专业会员,主要研究领域为服务计算,移动计算.