◆人工智能与无线网络◆

以 Google AMP 为例的网页加速技术研究*

陈贵豪,叶　进**,李　琳
(广西大学计算机与电子信息学院,广西南宁　530004)

摘要:网页的加载速度与用户体验紧密相关,决定着用户的去留。本研究以 Google AMP 网页加速技术为例,对网页加速技术展开研究,并借助其关键技术,对腾讯新闻、CNBC 新闻和 VOX 等网站进行改造,提出基于流行度分布和链路带宽的网页加载时间(Page Load Time,PLT)分析模型。结果表明,Google AMP 能有效降低网页加载时间,相对于原版网页,最高可降低 4.17%,即使在总体网络环境为最小往返时间(Minimum Round Trip Time,min_RTT)和丢包率较高、网络带宽较低的情况下,网页加载时间最高仍能降低 92.46%;同时在启用流行度缓存机制的情况下,最高可再降低 44%。本研究对进一步研究其他网页加速技术具有参考价值。

关键词:AMP　网页加载时间　Mahimahi　Selenium　流行度

中图分类号:TP393.092　　**文献标识码**:A　　**文章编号**:1005-9164(2021)03-0265-07
DOI:10.13656/j.cnki.gxkx.20210830.008

0　引言

当前,移动互联网进入高速发展的阶段。研究报告指出[1],截至 2018 年 12 月,中国 PC 互联网月度覆盖人数达 5.09 亿人,同比下降 3.9%;而移动互联网月独立设备数达到 13.47 亿台,同比增长 12.8%,移动端网民单日使用时长已达到 186 min,超过 PC 端的 2 倍。移动互联网用户在享受高带宽、无上限数据流量带来便利的同时,对低延迟、高响应的需求日益增长。对于移动互联网开发人员而言,产品的加载性能将决定着用户的去留,并且已经成为与同类产品竞争的焦点。

降低网页加载时间(Page Load Time,PLT),尤其是用户可感知的网页加载时间,是众多网页加速技术研究的关键问题。Galletta 等[2]研究网页延迟对用户体验的影响,认为当网页的显示时间越长,用户对网站的兴趣逐渐降低甚至可能会放弃该网页。

现有网页加速技术研究主要从网络传输协议、网络拓扑结构和网页资源调度机制 3 个层面进行优化。①网络传输协议方面:对传统 HTTP 协议改进和开发 SPDY[3]、HTTP/2[4]和 QUIC[5]协议等,以降低网络传输的时延,难点在于需要对浏览器内核的资源调

收稿日期:2020-12-07
*中国教育和科研计算机网 CERNET 赛尔网络下一代互联网技术创新项目(NGII20180305)资助。
【作者简介】
陈贵豪(1995-),男,硕士研究生,主要从事计算机网络研究以及数据中心网络协议优化研究。
【**通信作者】
叶　进(1970-),女,博士生导师,主要从事网络理论与实践、计算机软件、大数据领域研究,E-mail:yejin@gxu.edu.cn。
【引用本文】
陈贵豪,叶进,李琳.以 Google AMP 为例的网页加速技术研究[J].广西科学,2021,28(3):265-271.
CHEN G H,YE J,LI L.Research on Web Page Acceleration Technology: A Case Study of Google AMP [J].Guangxi Sciences,2021,28(3):265-271.

度机制进行重新更改和编译,对于普通网页开发者而言,难度极大。②网络拓扑结构优化方面:采用增加网络边缘缓存机制,Vakali 等[6]提出在靠近用户的范围内部署内容分发网络(Content Delivery Network,CDN),对用户群体密集访问的网站静态资源进行缓存,提升资源的命中率,但需要网站的所有者花费巨额资金购置或租用大量服务器。③网页资源调度优化方面:Wang 等[7]提出对页面资源进行依赖分析,对页面显示急需的资源优先加载。Netravali 等[8]提出网页加载时间取决于复杂的网页资源依赖,要降低网页加载时间,则需要追踪网页资源的深层次依赖关系。例如传统网页在加载过程中,需重点解决 Document Object Model (DOM)结构树构建被阻塞的问题,尽可能使非必需资源异步加载。

现有研究还包括客户端优化、基于代理的加速等。客户端优化从内容预取、预渲染和推测加载来降低网络高延迟对用户体验的影响,Netravali 等[9]通过在 Web 服务器预先计算页面的 JavaScript 堆和 DOM 树降低页面加载时间。基于代理的加速则是从划分客户端和远程代理服务器之间的负载过程来提高网络性能。Amazon[10]建立了客户端与 Amazon 云之间的处理架构,以提供用户更快速的移动浏览体验。Shaghayegh 等[11]针对 App-like 类的网页,提出代理服务器通过分析网页结构布局静态资源的相似性,提取并预先传输相对固定的静态资源,客户端仅需使用较少的流量和带宽即可填充动态数据资源,提升加载效率。

Google AMP (Google Accelerated Mobile Pages,以下简称 AMP)网页加速技术属于其中的网页资源调度优化方面的改进措施。本研究以 AMP 网页加速技术为例,对网页加速技术进行研究,并对 AMP 的网页实例进行性能测试与分析,比较在模拟高时延、高丢包率和低带宽的环境下,经过改造的 AMP 网页相对于原始网页的加载时间的差异,同时分析探讨不同流行度缓存机制下,对 AMP 网页加载时间起到决定性作用的因素。

1 AMP 设计与分析

AMP 是 Google 公司于 2016 年发起的一项开源代码库计划,目的是改进以 Google 全球搜索为中心的内容生态系统[12]。

Google 全球搜索对符合 AMP 开发标准的网页进行自动抓取,并将其缓存在 Google 的 CDN。AMP 网页在移动设备上的浏览器上以 60 帧的速度平稳加载,用户通过 Google 搜索到对应网站的 AMP 网页,就可以体验网页的极速加载。AMP 的生态系统包括超过 2 500 万个网站、100 多个技术提供商以及各大主流平台,涵盖出版、广告、电子商务及小型企业等领域[13]。

1.1 AMP 设计

AMP 由以下 3 个核心部件构成[14],分别承担网页构建、资源调度和内容缓存的任务。

①AMP HTML。这是使用 AMP 技术进行改造过的网页,其本质上是在原始 HTML 的基础上,使用自定义 AMP 组件拓展而成的 HTML 文件。经过改造后的 AMP 网页,移除了大部分的外部网络资源依赖。

②AMP JS 库。为实现 AMP 最佳性能的调度工具,它负责资源加载并处理相关的 AMP 标记,最主要的优化措施是阻塞外部资源的同步加载,仅允许外部资源异步加载。采取了以下措施:一是仅允许异步 JavaScript 脚本资源执行;二是对外部媒体资源(如图片、广告等)必需显式标定尺寸、位置;三是将所有可执行的异步 JavaScript 脚本设置在浏览器加载的非关键路径下;四是设置资源加载的优先级。

③AMP Cache。一种基于代理的内容分发网络,基于 Google CDN 服务器集群,用于缓存符合 AMP 标准的 HTML 网页,搜索引擎会自动抓取、缓存和传输 AMP HTML 网页。

AMP 技术的优势在于,不需要对用户客户端的操作系统和浏览器内核进行修改,服务器端也无需改造,需要修改的只有前端开发者的网页源码,同时结合并兼容了已有的 CDN 服务,从网页资源调度和网络拓扑结构两方面降低网页加载时间。

1.2 基于 AMP 的 PLT 分析

网页加载时间 PLT 是衡量网页加载性能的重要指标。影响网页加载时间的网络环境因素一般有最小往返时间(Minimum Round Trip Time, min_RTT,以下简称最小 RTT)、丢包率和网络带宽 3 个因素[15]。网页加载时间的计算公式如式(1)所示。

$$PLT = min\_RTT + \left(\frac{B}{C}\right) + T, \quad (1)$$

式中:第 1 项 min_RTT 表示从浏览器发起请求至接收到第 1 个 HTTP 响应的第 1 个字节的最小 RTT(忽略服务器的响应与处理时间)。第 2 项表示在一个确定带宽的链路上将网页所有对象回传所需要的

最少时间,其中 B 为网页大小,单位为 Mbit;C 为链路带宽,单位为 Mbps。第 3 项 T 表示浏览器将所有网页页面渲染,且所有异步资源完成加载所需的时间,即浏览器处理时间。

网页加载时间与流行度缓存机制的应用相关,本研究将网页加载时间与网页流行度进行数学分析。流行度(Popularity)是用于描述确定时间内网站中网页被访问的频数及其访问排名的关系的概率分布。将网页作为测试样本集合 P,集合 P 内的网页访问排名和访问频数满足幂率 Zipf 概率模型[16,17]。该幂律模型能够精确描述短时间内网页访问的流行度,其服从概率质量函数。式(2)给出了基于流行度的概率质量函数。

$$f(k;s;N)=\frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}, \tag{2}$$

式中:N 是网页组数;k 是相应网页的频数排名;s 是网页访问的流行度,s 越大表征用户访问集合中的网页较为集中,反之,越小则比较平均或分散。在反向代理服务器中设置固定的有限缓存空间和替换算法,以构造具有网页访问流行度的测试样本集合。

访问网页存在 2 种情况:①该网页在反向代理服务器缓存空间中;②该网页不在反向代理服务器中,需要到源服务器继续查找。第 1 种情况的平均查找时间设置为 α,第 2 种情况的平均查找时间设置为 β。网页在反向代理服务器查找到的概率与该网页的幂律模型和该网页的频数排名有关,因此式(1)中的 min_RTT 可定义为

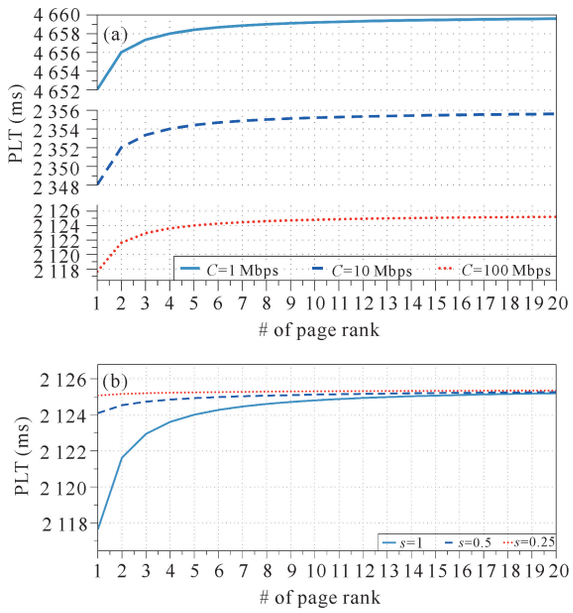
$$\text{min\_RTT}=f(k)\times\alpha+(1-f(k))\times\beta。 \tag{3}$$

结合式(2)可得出式(4),即基于流行度的 PLT 计算式:

$$\text{PLT}=f(k)\times\alpha+(1-f(k))\times\beta+(B/C)+T。 \tag{4}$$

由式(4),以 $N=300$ 为例,图 1 给出截取网页频数排名前 20 的网页,展示网页加载时间 PLT 分别关于流行度 s、链路带宽 C 的函数关系分析。

需要特别指出的是,浏览器处理时间 T 与软硬件性能相关,因此只做链路带宽 C 与流行度 s 的定量分析。在此,网页大小 B 和浏览器处理时间 T 均取平均值,分别为 2.56 Mbit 和 2 s,网页在服务器查找时间 α、β 分别取 0.05,0.1 s。图 1a 中流行度 $s=1$,图 1b 中链路带宽 $C=100$ Mbps。



(a) 链路带宽 C;(b) 流行度 s

(a) Link bandwidth C;(b) Popularity s

图 1 链路带宽 C 和流行度 s 分别与网页加载时间 PLT 的关系分析

Fig. 1 Analysis of the relationship between link bandwidth C and popularity s and page load time (PLT)

2 验证实验

2.1 测试环境的网络拓扑结构

如图 2 所示,使用 3 台服务器主机与 1 台客户机在局域网内搭建测试环境,使用一个交换带宽为 300 Mbps 的路由器有线连接组网,从任意主机到路由器的平均 ping 值均小于 1 ms,用于模拟真实环境下,用户通过反向代理服务器访问网站源服务器的应用场景。其中:源服务器 1,2 均用于部署抓取到本地的网页资源,使用 Apache 作为 Web 服务器软件[18]。2 台服务器网页资源自动同步,且以源服务器1为参

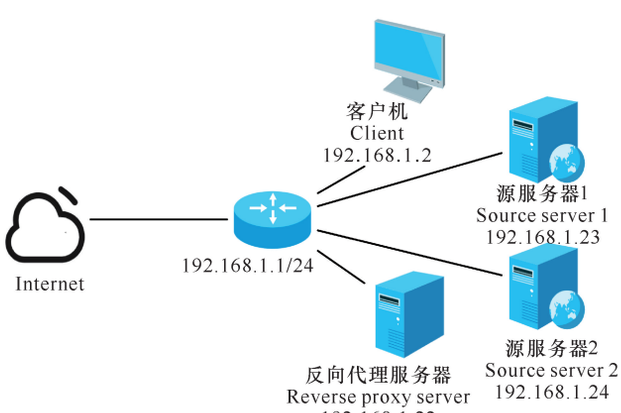


图 2 测试环境网络拓扑结构图

Fig. 2 Network topology structure diagram of test environment

照。反向代理服务器使用 Nginx 作为 Web 服务器软件[19],使用权重相同的轮询方式依次访问源服务器 1,2。客户机在局域网内部通过反向代理服务器请求并获取网页资源。

2.2 交叉对比实验

设计 9 组交叉对比实验,分别测试在不同网络环境下(表 1 给出网络环境参数配置)腾讯新闻、CNBC 新闻和 VOX 新闻 3 个网站的 AMP 网页(AMP Page)和原版网页(Canonical page)的网页加载时间差异。每组测试在 3 个网站中各随机选取相同的 100 组网页(包括 AMP 网页和原版网页),共 300 组网页。

表 1 测试环境参数配置

Table 1 Parameter configuration of test environment

组别 Group	min_RTT (ms)	丢包率 Loss rate (%)	带宽 Bandwidth (Mbps)
min_RTT 实验组 min_RTT experiment group	{50,100,200}	0	100
上/下行丢包率实验组 Upload/download packet loss rate group	100	{0,10,20}	100
带宽实验组 Bandwidth experiment group	10	10	{100,10,1}

2.3 自动化性能测试

针对上述 3 个网站结构进行分析,设计并实现了分别针对 3 个网站的、基于 Python 语言及 pyquery 框架的脚本,用于解析和抓取网页文档资源。为便于进行自动化性能测试,将同一页面的 AMP 网页和原版网页通过重命名进行区分。表 2 为 AMP 网页和原版网页在访问 URL 上,通过路径/canonical/和/amp/进行区分,其中[NEWS]的枚举值集合为{qqnews,cnbc,vox}。

表 2 同一页面的 AMP 网页和原版网页的 URL

Table 2 The URL of the AMP page and the canonical page on the same page

网页版本 Version	URL
AMP 网页 AMP page	192. 168. 1. 22/[NEWS]/amp/example. html
原版网页 Canonical page	192. 168. 1. 22/[NEWS]/canonical/example. html

使用 Selenium 自动化测试套件、Mahimahi 网络环境模拟工具、Trickle 带宽控制工具和 Firefox 浏览器进行组合测试。Selenium 是为浏览器开发者或网页开发者提供的自动化测试套件[20]。Mahimahi 是由 MIT 的 Netravali 开发并在 USENIX 会议上提出的,用于模拟网络环境并记录 HTTP 协议请求过程的工具[15]。Trickle 是由 Google 的 Marius 开发并在 USENIX 会议上提出的,用于在 Unix/类 Unix 系统上对网络带宽进行控制和整形的工具[21]。Firefox 浏览器是由 Mozilla 基金会支持的开源项目[22]。

图 3 给出自动化性能测试流程。首先使用 Trickle 限制系统的网络带宽,Mahimahi 用于限制系统网络请求的最小 RTT 和丢包率;然后自动化脚本将自动读取并调用 Selenium 控制浏览器批量加载网页;最后 Firefox 根据脚本的调用加载指定的 AMP 网页和原版网页。

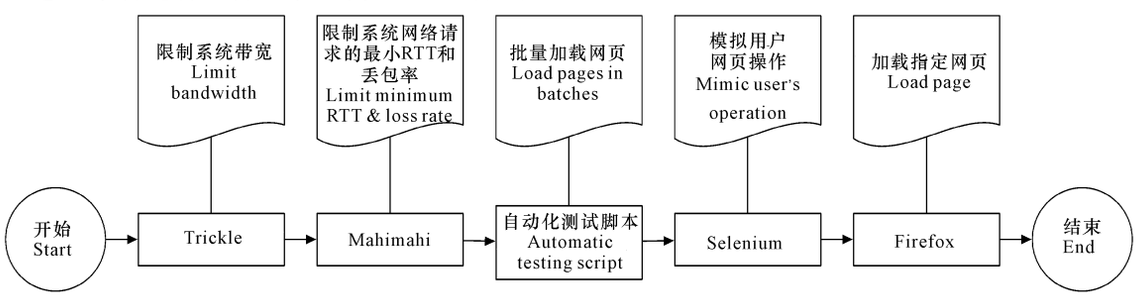


图 3 自动化性能测试流程图

Fig. 3 Test flow chart of automation performance

设计并实现自动化测试脚本,用于自动获取网页资源地址,使用 Selenium Driver 提供的系统接口,调用 Firefox 官方提供的驱动程序 Geckodriver,远程控制 Firefox 浏览器加载指定网页,并通过控制台模块执行 JavaScript 脚本,读取 JavaScript 运行环境的堆栈信息。

Firefox 浏览器的设计严格遵循 W3C 提出的相关标准。Firefox 提供了在 JavaScript 运行环境内调取网页加载时间点的接口 window. Performance. timing[23]。本文定义的网页加载时间可根据计算 domContentLoadedEventEnd 时间点和 navigationStart 时间点之差得到。

自动化测试脚本在每次运行前需要将 Firefox 浏览器的缓存和 Cookie 彻底清空,同时需要将 Firefox 设置为“每次关闭 Firefox 时删除 Cookie 与网站数据”。脚本在每次执行打开网页任务时,均调用 browser.delete_all_cookies(),以防浏览器再次访问相同网页时直接读取浏览器内部缓存,导致网页加载时间的记录失真。

3 验证结果

3.1 不同网络环境下 AMP 网页加载时间变化

在搭建的实验环境中,本文在表 2 给出的网络环境配置参数下,对 3 个网站的 AMP 网页和原版网页分组(Page Groups)进行网页加载测试,取中位数值和占总数 95%的网页加载时间值。图 4 为 AMP 网页加载时间相对于原版网页的降低比例曲线(Plt reduction ratio)。

图 4 中,min_RTT 为 50 ms 时,PLT 降低比例最高的是 88.61%;Loss 为 20%时,PLT 降低比例最高的是 94.17%;Bandwidth 为 1 Mbps 时,PLT 降低比例最高的是 92.46%。结果表明,最小 RTT 的升高或丢包率的升高,都将导致 AMP 网页加载时间总体逐步上升,AMP 网页的性能表现好,加载时间可容忍,而网络带宽的降低对 AMP 网页加载时间总体影响有限。

3.2 使用与不使用网页流行度缓存机制下,AMP 网页加载时间变化

在 min_RTT 为 200 ms,丢包率为 0 和带宽为 100 Mbps 的网络环境中,分别设置不使用流行度(w/o s)与使用流行度 $s=\{1,0.5,0.25\}$ 的缓存机制,进行 300 组网页加载测试,并得到 AMP 网页加载时间的累积分布曲线(Cumulative Distribution Function,CDF),如图 5 所示。结果表明,在相同网络环境中,流行度缓存机制可以使 AMP 网页加载时间最高降低 44%。

获得上述结果主要归结于以下 3 个方面:

①在网页结构方面。AMP HTML 内的 CSS 资源完全内联于文档内部,对图片等媒体资源需要标定尺寸。内联的 JavaScript 脚本不可同步执行,对于外部链接的 JavaScript 资源必须使用 async 属性显式标定。从本次实验的所有网页的结构上看,渲染和显示资源大部分内联于网页文档之内,腾讯新闻、CNBC 新闻和 VOX 新闻的 AMP 网页相比于原版网页需要发起的网络请求更少,因此在 min_RTT、丢包率和网络带宽方面均占有优势。

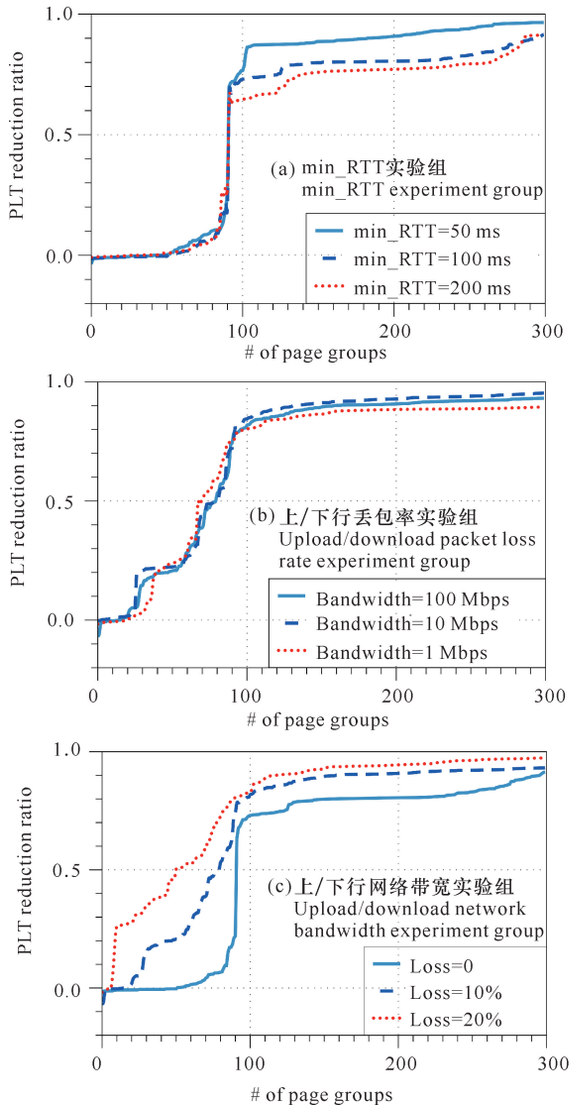


图 4 不同网络环境下 AMP 网页加载时间降低比例

Fig. 4 PLT reduction rate of AMP pages under different network environments

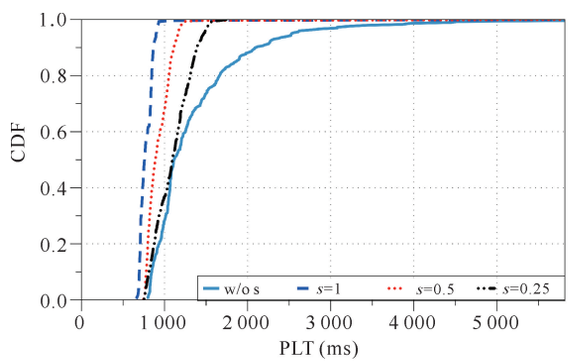


图 5 流行度缓存机制下 AMP 网页加载时间的累积分布曲线

Fig. 5 Cumulative distribution curve of AMP page load time under the popularity cache mechanism

②在资源调度机制方面。AMP JS 在网页渲染加载期间,阻止浏览器按照资源在页面内的顺序排布次序进行加载,优先对 CSS 结构树进行渲染,优先构建 DOM 结构树,阻断任何将要阻塞 DOM 结构树构建的 JavaScript 脚本的解析。3 种新闻网站原版页面内的资源排布比较杂乱,除腾讯新闻的原版网页在资源排布上比较接近 AMP 网页之外,内联的 JavaScript 脚本可以随时阻塞 DOM 的构建。相比之下,AMP 网页的资源调度机制更优。

③在流行度缓存机制方面。具有流行度缓存机制的代理服务器通过缓存网页访问的相关资源,提高网页资源在代理缓存中的命中率,从网络传输方面进一步降低传输时间。同时也印证了 AMP 需要依靠 Google 的 CDN 服务器集群中运行的 AMP Cache,这对于提升网络资源传输效率具有十分重要的作用。

4 结论

本文通过对网页加载时间和用户体验的研究,对 Google AMP 网页加速技术进行分析、研究和自主设计实验。实验结果表明,AMP 相对于原版网页,最高可降低 94.17%;即使在总体网络环境为最小往返时间和丢包率较高、网络带较低的恶劣情况下,网页加载时间最高仍能降低 92.46%,同时在启用流行度缓存机制的情况下,最高可再降低 44%。然而,AMP 性能提升的代价和成本要转移到网页开发者,开发复杂度的上升和开发周期的延长,意味着 AMP 在开发架构上仍有改进和提升空间。AMP 与 Google 最新推出的应用层传输协议 QUIC 结合使用,将有助于进一步提升其性能[24],也是下一步的研究方向。

参考文献

[1] 中国互联网流量年度数据报告 2018 年[C]//艾瑞咨询系列研究报告(2019 年第 2 期).上海:上海艾瑞市场咨询有限公司,2019:67-177.

[2] GALLETTA D F,HENRY R,MCCOY S,et al. Web site delays:How tolerant are users? [J]. Journal of the Association for Information Systems,2004,5(1):1-28.

[3] WANG X S,BALASUBRAMANIAN A,KRISHNAMURTHY A,et al. How speedy is SPDY [M/OL]// 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI14). Seattle WA,USA:USENIX Association,2014:387-399. [2020-12-07]. https://www. usenix. org/system/files/conference/nsdi14/nsdi14-paper-wang_xiao_sophia. pdf.

[4] Internet Engineering Task Force. Hypertext transfer protocol version 2 [R/OL]. (2015-05-01) [2020-12-07]. https://tools. ietf. org/html/rfc7540.

[5] CUI Y,LI T,LIU C,et al. Innovating transport with QUIC:Design approaches and research challenges [J]. IEEE Internet Computing,2017,21(2):72-76.

[6] VAKALI A,PALLIS G. Content delivery networks:Status and trends [J]. IEEE Internet Computing,2003,7(6):68-74.

[7] WANG X S,KRISHNAMURTHY A,WETHERALL D. Speeding up web page loads with shandian [M/OL]//Proceeding of the 13th USENIX Symposium on Networked Systems Design and Implementation. Berkeley,CA,USA:USENIX Association,2016:109-122. [2020-12-07]. https://dl. acm. org/doi/abs/10. 5555/2930611. 2930619.

[8] NETRAVALI R,GOYAL A,MICKENS J,et al. Polaris:Faster page loads using fine-grained dependency tracking [M/OL]// Proceeding of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI16). Berkeley,CA,USA:USENIX Association,2016:123-136. [2020-12-07]. https://dl. acm. org/doi/10. 5555/2930611. 2930620.

[9] NETRAVALI R,MICKENS J. Prophecy:Accelerating mobile page loads using final-state write logs [C]// USENIX Association. Proceeding of 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI18). Renton,WA,USA:USENIX Association,2018:249-266. [2020-12-07]. http://web. cs. ucla. edu/~ravi/publications/prophecy_nsdi18. pdf.

[10] AMAZON. What is Amazon Silk? [EB/OL]. (2019-06-07) [2020-12-07]. https://www. computerhope. com/jargon/a/amazon-silk. htm.

[11] SHAGHAYEGH M,MAYANK S,NETRAVALI R. Fawkes:Faster mobile page loads via app-inspired static templating [M/OL]//Proceeding of 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI' 20). Santa Clara,CA,USA:USENIX Association,2020:879-894. [2020-12-07]. https://www. usenix. org/system/files/nsdi20-paper-mardani. pdf.

[12] GOOGLE. AMP is a web component framework to easily create user-first websites [EB/OL]. (2019-6-15) [2020-12-07]. https://www. ampproject. org/zh_cn/.

[13] GOOGLE. E-Commerce AMP camp [EB/OL]. [2020-12-07]. https://www. ampproject. org/learn/showcases/.

[14] GOOGLE. How AMP works [EB/OL]. (2017-03-14) [2020-12-07]. https://www.ampproject.org/zh_cn/learn/overview/.

[15] NETRAVALI R, SIVARAMAN A, DAS S, et al. Mahimahi: Accurate record-and-replay for HTTP [M/OL]// Proceeding of the 2015 USENIX Annual Technical Conference (USENIC ATC'15). Santa Clara, CA, USA: USENIX Association, 2015: 417-429. [2020-12-07]. https://www.usenix.org/system/files/conference/atc15/atc15-paper-netravali.pdf.

[16] PASCHOS G S, IOSIFIDIS G, TAO M X, et al. The role of caching in future communication systems and networks [J]. IEEE Journal on Selected Areas in Communications, 2018, 36(6): 1111-1125.

[17] BRESLAU L, CAO P, FAN L, et al. Web caching and Zipf-like distributions: Evidence and implications [M/OL]// Proceeding of 18th IEEE Conference on Computer Communications (IEEE INFOCOM'99). New York, NY, USA: IEEE, 1999: 126-134. [2020-12-07]. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=749260.

[18] APACHE. HTTP server project [EB/OL]. [2020-12-07]. http://httpd.apache.org.

[19] NGINX. Nginx news [EB/OL]. [2020-12-07]. http://nginx.org.

[20] Selenium. What is Selenium [EB/OL]. (2019-07-06) [2020-12-07]. https://docs.seleniumhq.org.

[21] ERIKSEN M A. Trickle: A userland bandwidth shaper for Unix-like systems [M/OL]// Proceeding of 2005 USENIX Annual Technical Conference. Marriott Anaheim, CA, USA: USENIX Association, 2005: 61-70. [2020-12-07]. https://static.usenix.org/event/usenix05/tech/freenix/full_papers/eriksen/eriksen.pdf.

[22] Mozilla Developer Network. Mozilla makes browsers, apps, code and tools that put people before profit [EB/OL]. [2020-12-07]. https://www.mozilla.org/zh-CN/about/.

[23] Mozilla Developer Networks. Performance [EB/OL]. [2020-12-07]. https://developer.mozilla.org/zh-CN/docs/Web/API/Performance.

[24] JUN B, BUSTAMANTE F E, WHANG S Y, et al. AMP up your mobile web experience: Characterizing the impact of Google's accelerated mobile project [C]//The 25th Annual International Conference on Mobile Computing and Networking (MobiCom'19). New York, NY, USA: ACM, 2019: 1-14. https://doi.org/10.1145/3300061.3300137.

Research on Web Page Acceleration Technology: A Case Study of Google AMP

CHEN Guihao, YE Jin, LI Lin

(School of Computer, Eletronics and Information, Guangxi University, Nanning, Guangxi, 530004, China)

Abstract: The loading speed of the web page is closely related to the user's experience, which determines the user's staying and leaving. In this study, Google AMP was taken as an example to study the page acceleration technology. With the help of its key technologies, Tencent News, CNBC News, VOX and other websites were reformed, and a Page Load Time (PLT) analysis model based on popularity distribution and link bandwidth was proposed. The results show that compared with the original webpage, Google AMP can effectively reduce the loading time of the webpage by up to 94.17%. Even with the overall network environment of high minimum RTT, high loss rate of the packet, and low network bandwidth, page load time can still be reduced by 92.46%. At the same time, when the popularity caching mechanism is enabled, AMP can be reduced by up to 44%. This study has a reference value for further research on other web acceleration technologies.

Key words: AMP, page loading time, Mahimahi, Selenium, popularity

责任编辑:陆 雁