

前端性能优化4大环节最佳实践与研发流程

技术漫谈 2022-01-19 08:32

以下文章来源于Lark FE，作者李高禄



Lark FE

暂无描述



大型项目发展到一个阶段都绕不开性能优化，高性能是高品质产品的重要特征。如果你是研发同学，你会关心如何入手优化，优化的链路都有哪些环节；如果你是Team Leader，你一定很关心用什么易于执行，易于验证的研发流程，来指导前端团队交付高性能的产品，并确保产品的性能始终是健康的、可持续的。

本文将介绍前端优化链路的全景图和可持续的性能优先研发流程。

性能指标

性能优化的第一步是要定义好指标，指标能帮助我们衡量优化结果的好坏，是指导我们如何进一步优化的标准。

如何定义指标

有几个途径可以定义出合理的指标：

对齐业界标准指标：适合一些常见的指标，比如 $TTI < 4s$ 。对齐竞品的指标：竞品是最好的老师，调研发现我们和竞品的差距，优化到竞品的标准就是进步。根据业务的数据来定指标：当我们的业务比较特殊，或者我们需要更高的标准，而又没有竞品可以参考时，我们可以先分析上报数据，根据数据的平均值来求一个合理的值，比如 指标值=平均值/2。

常见指标

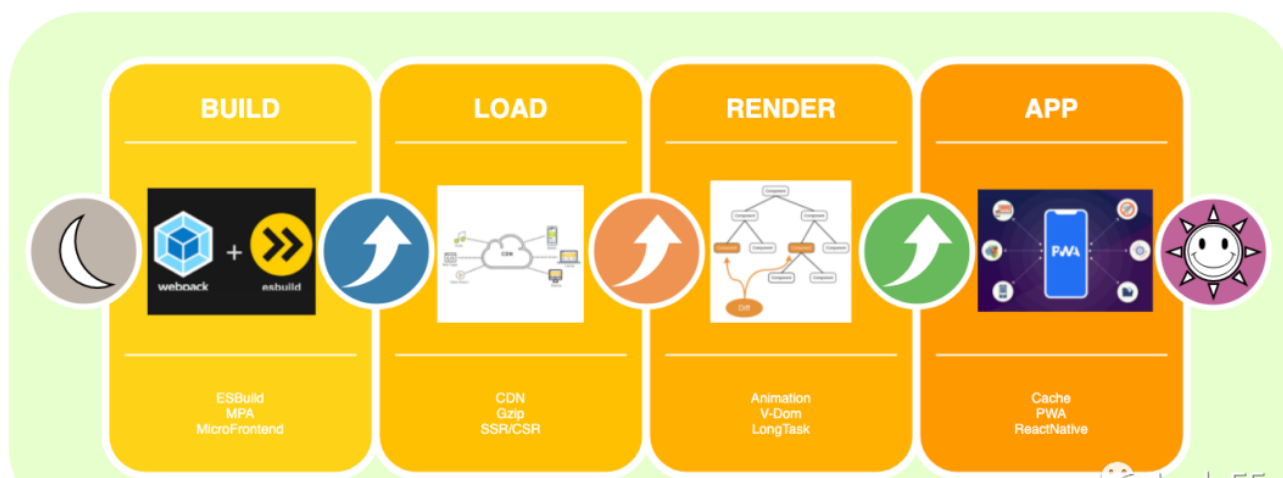
来自 web.dev

### 指标	符合预期	需优化	
快	中	慢	
First Contentful Paint	0-1.8s	1.8-3s	>3s
First Meaningful Paint	0-2s	2-4s	>4s
Speed Index	0-3.4s	3.4-5.8s	>5.8s
First CPU Idle	0-4.7s	4.8-6.5s	>6.5s
Time to Interactive	0-3.8	3.9-7.3s	>7.3s
Max Potential First Input Delay	0-130ms	130-250ms	>250ms
Total Blocking Time	0-200ms	200-600ms	600ms
Largest Contentful Paint	0-2.5s	2.5-4s	4s

前端如何计算并收集上报这些指标，是另一个话题了，可以关注我们的公众号，后续会发布《端体验优化之全链路监控》的相关文章。

优化链路

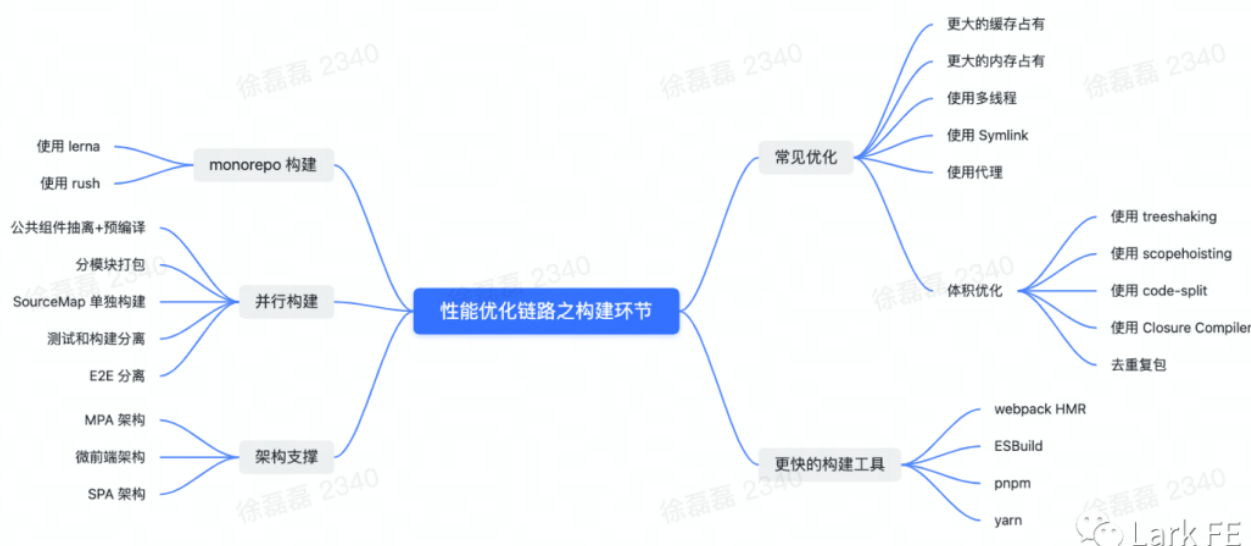
前端性能优化有很多维度，我们可以归类为几个环节：构建环节、网络环节、渲染环节、应用环节。当这几个环节实现全流程优化时，性能可以达到更优的表现。而每个环节都有各自的优化手段，下面我们就介绍这几个环节的优化最佳实践和案例。（篇幅有限，我们还有很多的优化案例比如 飞书前端项目打包时间优化，SSR，MPA专项，HTTP3升级等等，后续后会以单独的文章发出～）



构建环节优化

架构升级也是性能优化的重要手段，需要构建环节支撑，比如支持 SPA 构建，支持 MPA 构建，支持微前端构建。工欲善其事，必先利其器，优化构建时间缩短能大幅度提升研发体验，为快速验证性能优化结果做好准备。构建系统优化能提升研发效率，是个很大的话题，我们就提一些通用的思路。

最佳实践

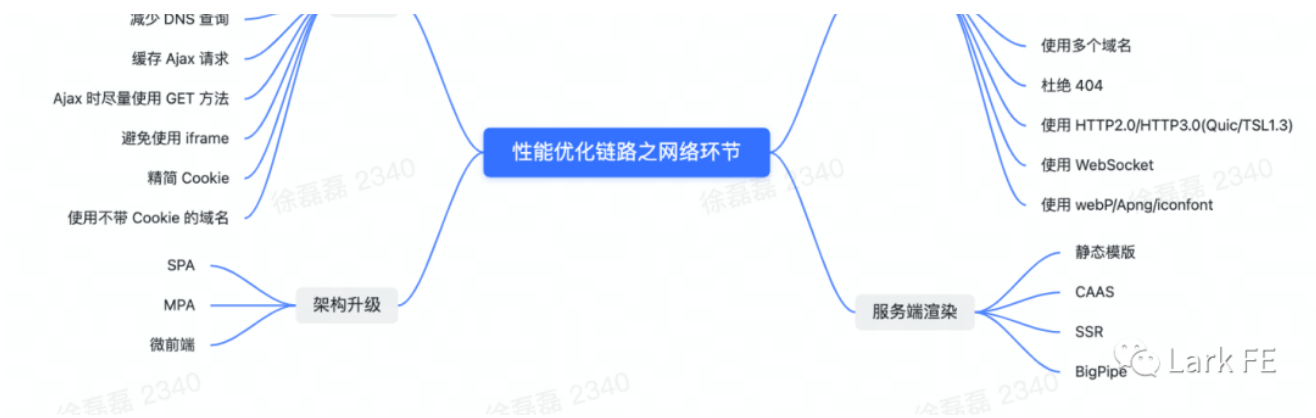


网络环节优化

通常网页加载超过4s还没看到东西，用户就感知到慢了，大于 10s 会放弃等待。网络优化若能升级网络协议，会得到质的提升，比如升级到 HTTP2.0 支持并行下载。Chrome 已经支持 HTTP3.0（默认关闭），底层基于 UDP，自家的网页应用直接享受 0-RTT。现在网页应用光加载静态资源的体积动不动就几MB到十几 MB，光靠升级服务和协议是不够的，前端也需要很多优化策略，比如采用服务端渲染、升级架构。

最佳实践





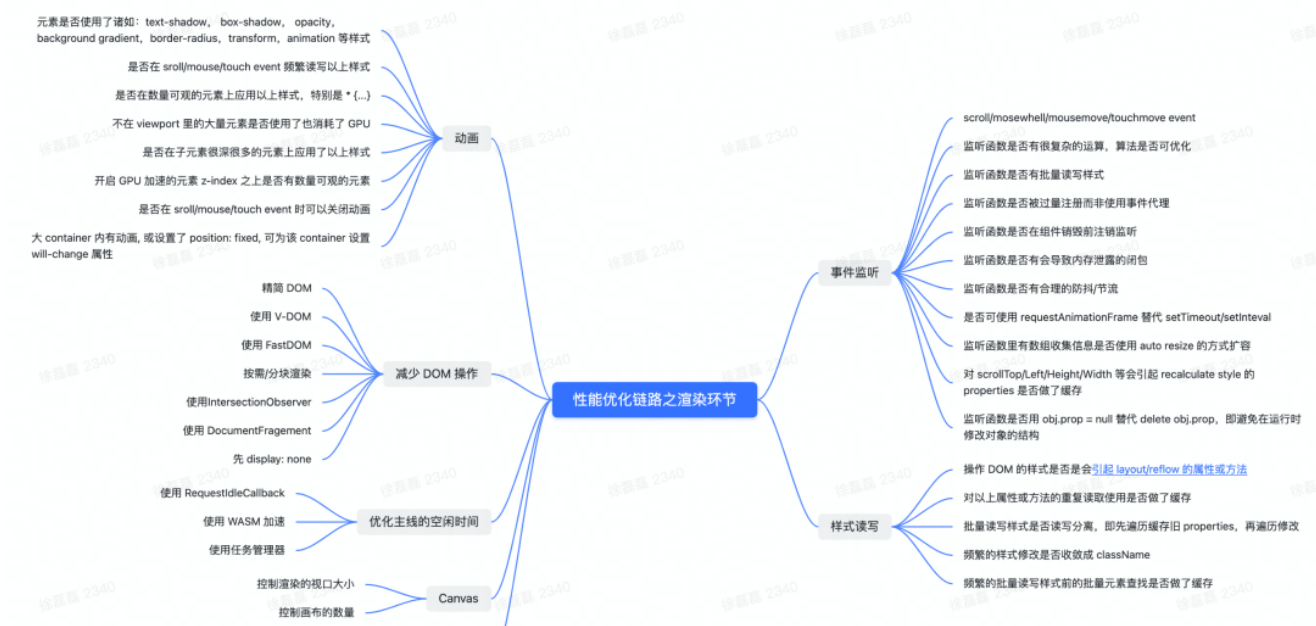
「性能优化链路之网络环节」

渲染环节优化

渲染优化保障是用户体验的流畅性的重要环节，而且渲染性能会影响内存和 CPU 的持续占有。通常我们认为的流畅是指帧数达到 60FPS 或稳定 30FPS，浏览器绘制一帧一般需要 10ms，用户在操作过程中有响应延时的时间不同用户体验也不同：

响应延时	用户感知
0 - 16ms	每帧 16ms，即 60FPS，用户感觉很流畅
0 - 100ms	用户操作页面（如点击一个按钮）100ms 之内响应，用户感觉到响应很及时
100 - 1000ms	用户操作页面没有感到有明显的卡顿，多数用户都可以顺利完成余下的下操作
>1s	大于1s用户感知到明显卡顿，但依然可以忍受，能坚持完成剩下的必须的操作
>10s	10s 真是太久了，用户失去耐心甚至放弃，多半会关网页走人，用户流失

最佳实践



「性能优化链路之渲染环节」

应用环节优化

再好的网络环境也比不上离线加载的速度，无论是加载速度还是交互流畅性，前端其实一直在追赶原生的体验。主要围绕着离线优化、热更和跨端渲染框架。

最佳实践



「性能优化链路之应用环节」

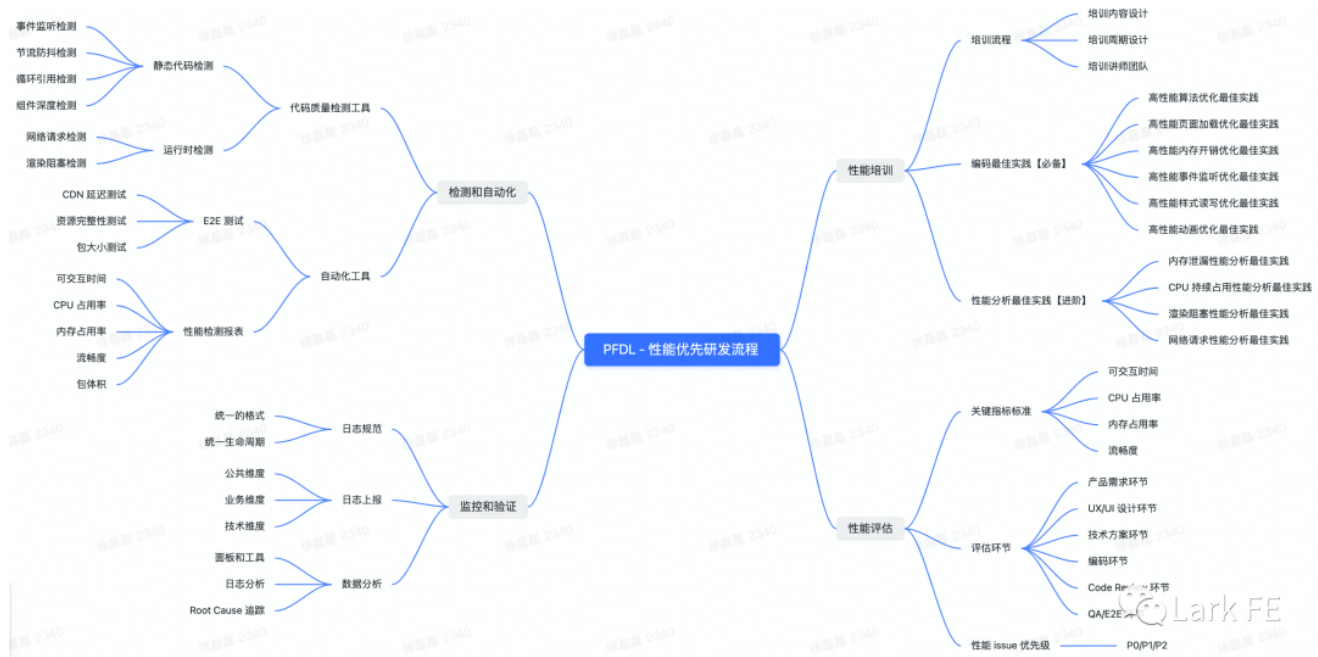
PFDL - 性能优先研发流程





性能优先研发流程（Performance First Development Lifecycle，简称 PFDL）。PFDL 专注于性能，以较小的成本影响现有研发流程，提升产品整体性能的水准，交付高质量的产品体验为目标。

PFDL 分为以下部分：新人培训，性能评估，检测和自动化，监控和验证。



「PFDL - 性能优先研发流程」

性能培训

开发团队不断有新人加入，性能优先必须深入人心，那么培训是必不可少的。培训的内容包括高性能编码最佳实践，性能分析最佳实践等。培训讲师，可来自团队经验丰富的研发人员，也可请业界大牛，甚至网络上优质付费/免费资源。培训将是周期性进行的，系统化的，内容覆盖全面的。

性能评估

以用户体验为中心，定义出关键指标标准，可以借鉴 Google 的 RAIL 模型/Web Vitals，对比竞品，现有产品的性能进行评估：可交互时间，CPU 占用率，内存占用率，流畅度等。应该在各个环节加入性能评估，包括产品需求环节，UX/UI 设计环节，技术方案环节，编码环节，Code Review 环节，QA/E2E 环节等。对性能 issue 需要优先级的评估：P0/P1/P2。

检测和自动化

我们需要更多工具，能帮助我们更高效检测出性能的问题，能自动优化代码的性能。我们需要网罗或设计开发这些工具，这些工具尽可能的做到相互关联，相互辅助，并可流程化，尽可能的自动化。

监控和验证

监控数据如何反应接近真实的用户体验始终是个难点，定义规范的日志，建设有效的面板，使用强大的工具，能帮助我们主动关注产品的质量提前采取优化行动，同时更有效的在海量日志中追踪性能问题 Root Cause，快速定位问题直到修复。

点击下方“**技术漫谈**”，选择“**设为星标**”
第一时间关注技术干货！



技术漫谈

有知识，有温度，有态度。技术漫谈，新一代技术人的聚集地！

91篇原创内容



公众号

喜欢此内容的人还喜欢

3元获得ChatGPT官网免费账号指南

AI语者



网站推荐——给大伙更新一波追剧网站

大猫小鼠



ChatGPT注册



