

基于 RISC-V 的Chromium 性能优化实践

李 扬

阿里巴巴达摩院 技术专家



目录

Contents

01 | 性能优化背景

02 | 性能优化实践

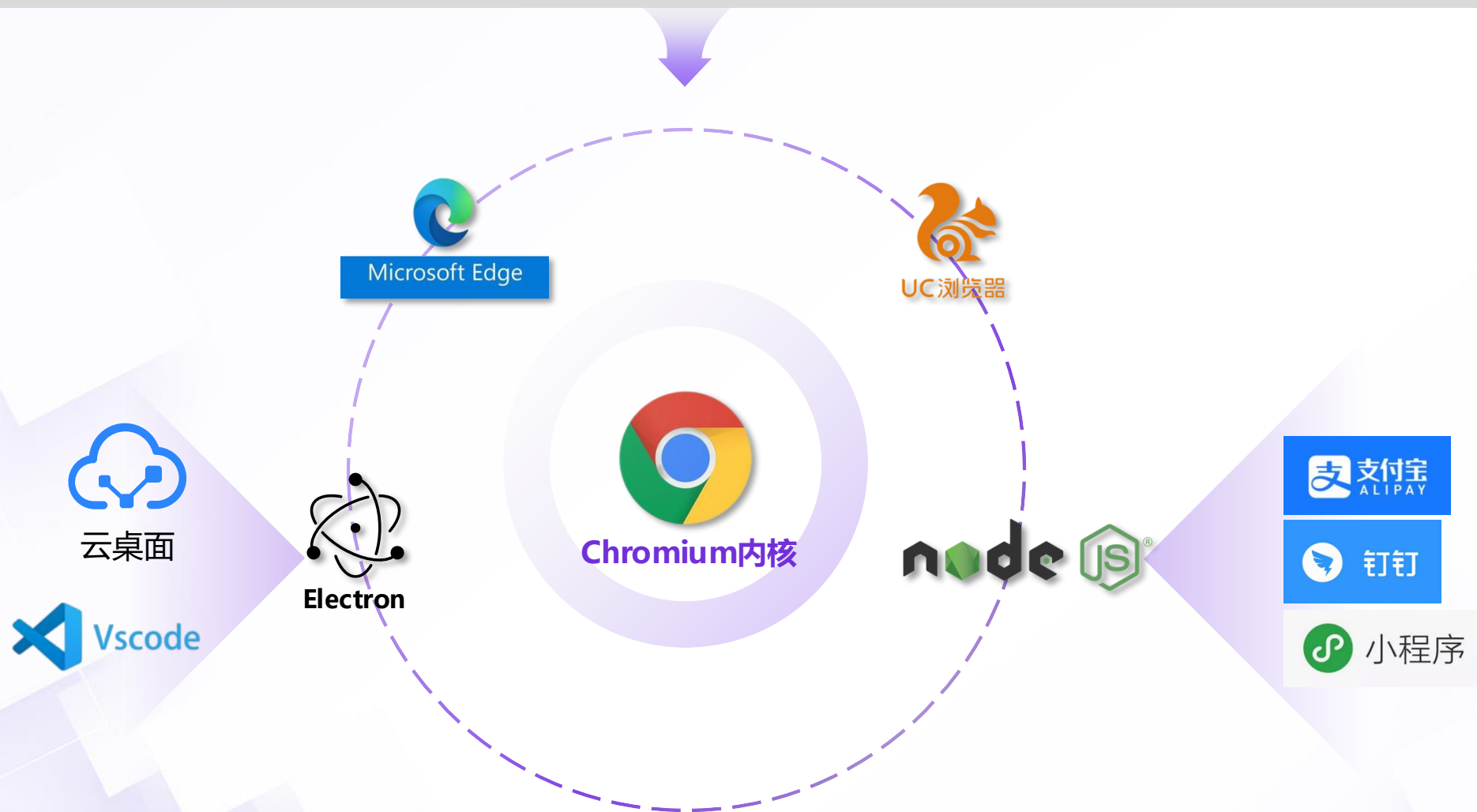
03 | 性能优化成果



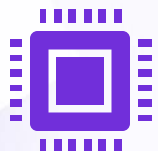
01 | 性能优化背景

优化背景

随着端云一体应用的不断发展，越来越多的PC端、移动端、云端应用基于**开源浏览器内核**而构建。



2018年开始有RISC-V提交，但未正式支持



版本现状

- 2018年开始有RISC-V提交，但未正式支持
- 版本构建尚有困难
- 性能难以满足产品化要求



性能问题

- 网页响应慢
- 云应用卡顿
- 用户交互速度慢
- 小程序响应速度慢
- 视频播放卡顿

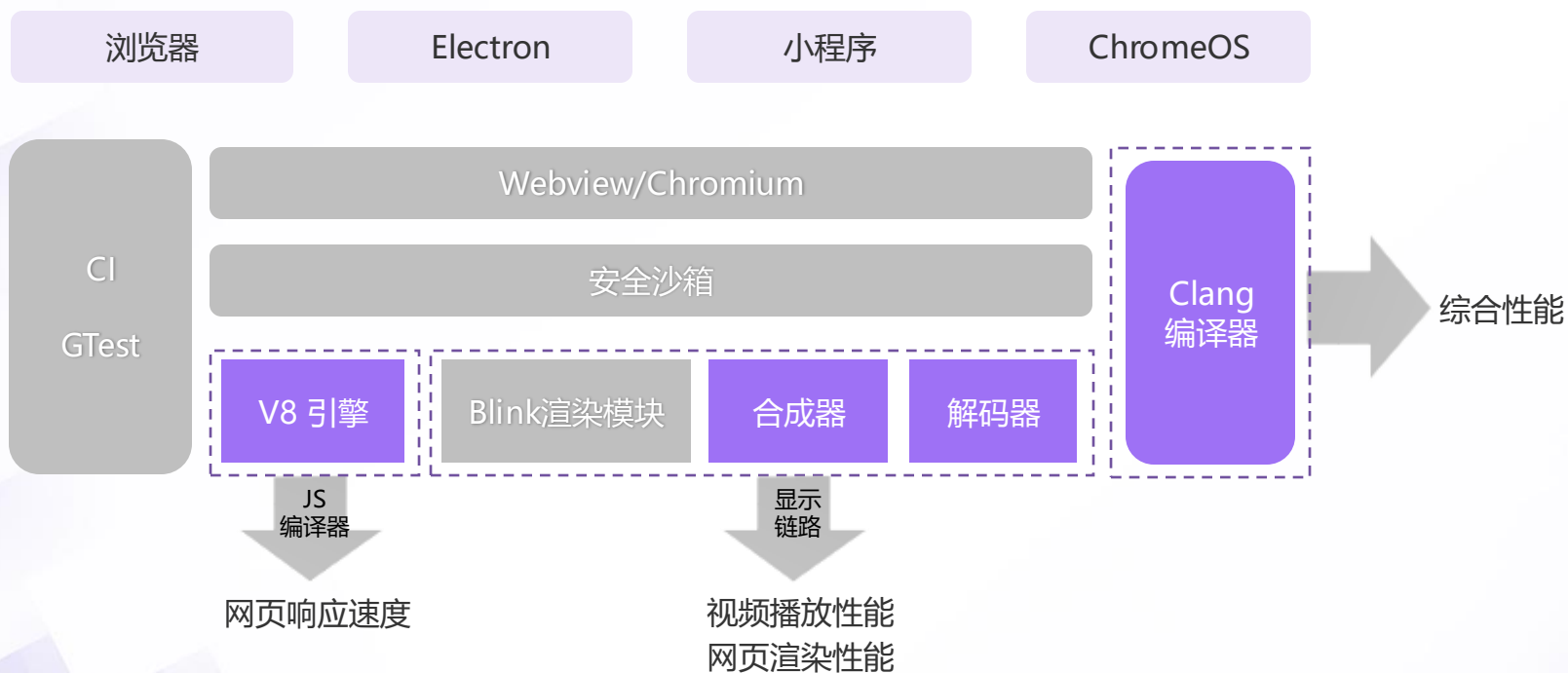




02 | 性能优化实践

优化地图

代码行数: 3000万+
三方库数量: 100+
涉及开发语言: C++/JavaScript/HTML/CSS/Python/Java/Rust

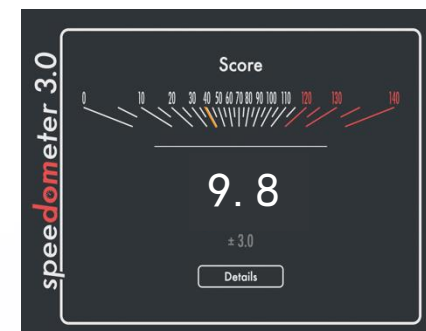
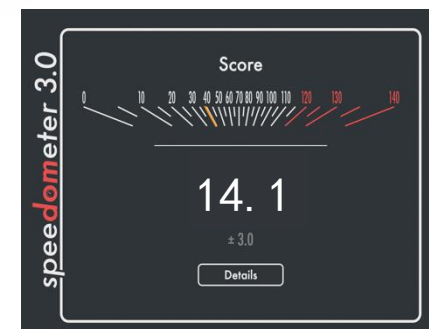


优化1 – 工具链迭代优化

优化方法

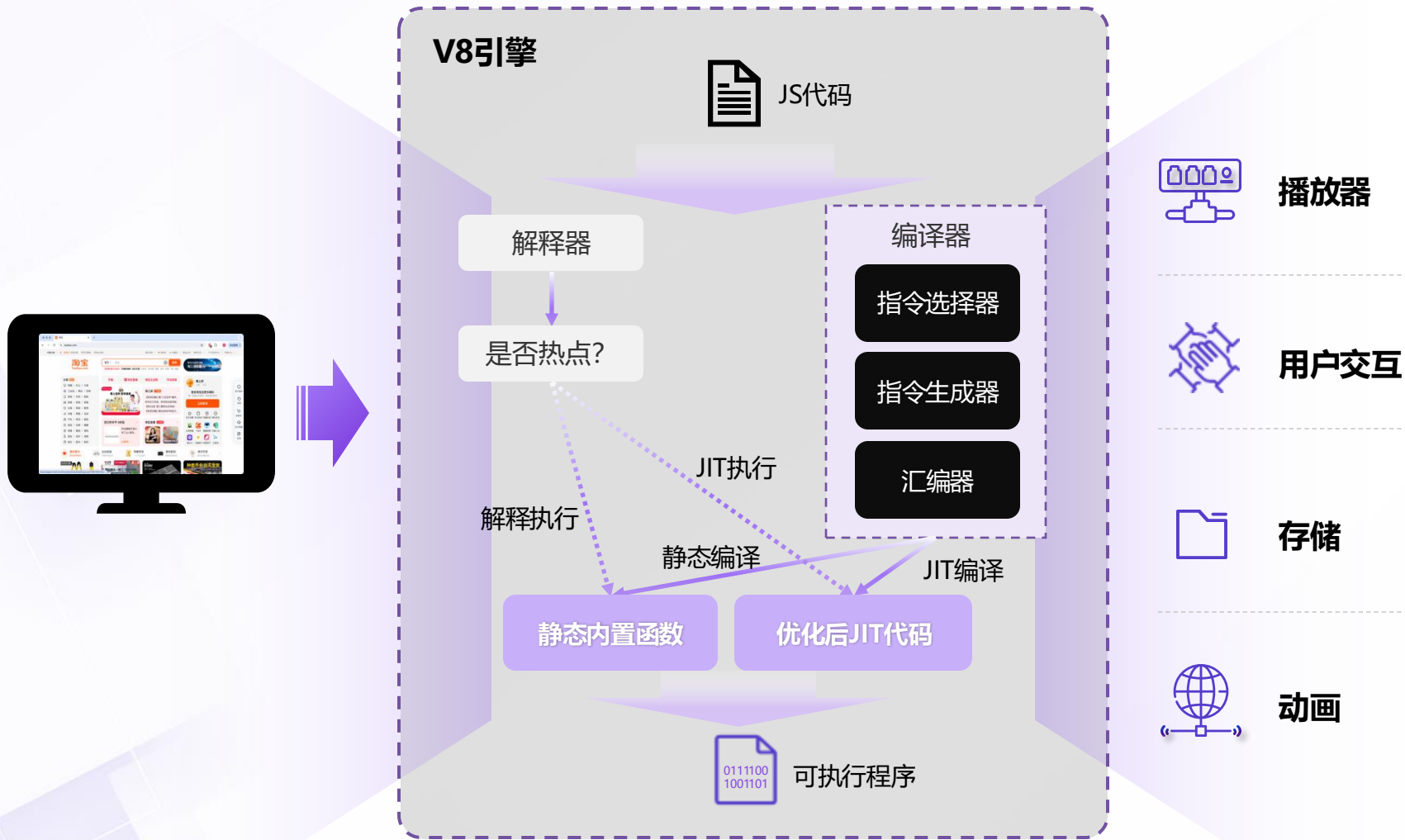


优化效果



Benchmark优化提升

优化2 – V8 JS引擎优化



优化2 – V8 JS引擎优化

优化方法

指令选择器

根据RV64的指令集特点，优化了指令选择和指令融合逻辑，使得在RV64架构上可以生成更简洁的中间代码表达

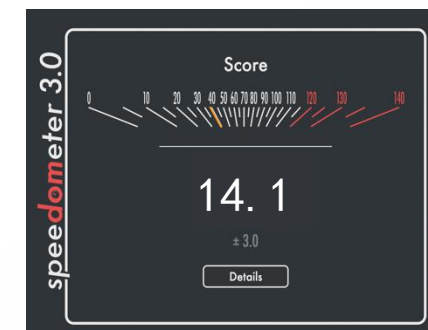
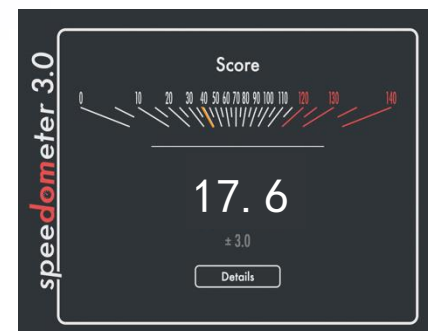
指令生成器/汇编器

丰富RV后端指令集，在宏汇编器、汇编器中加入玄铁扩展指令集支持，使得生成的二进制指令更加紧凑高效

静态内置函数

通过中间表达优化、手动汇编优化、跳板机制优化进一步优化内置函数，缩短翻译执行时间

优化效果

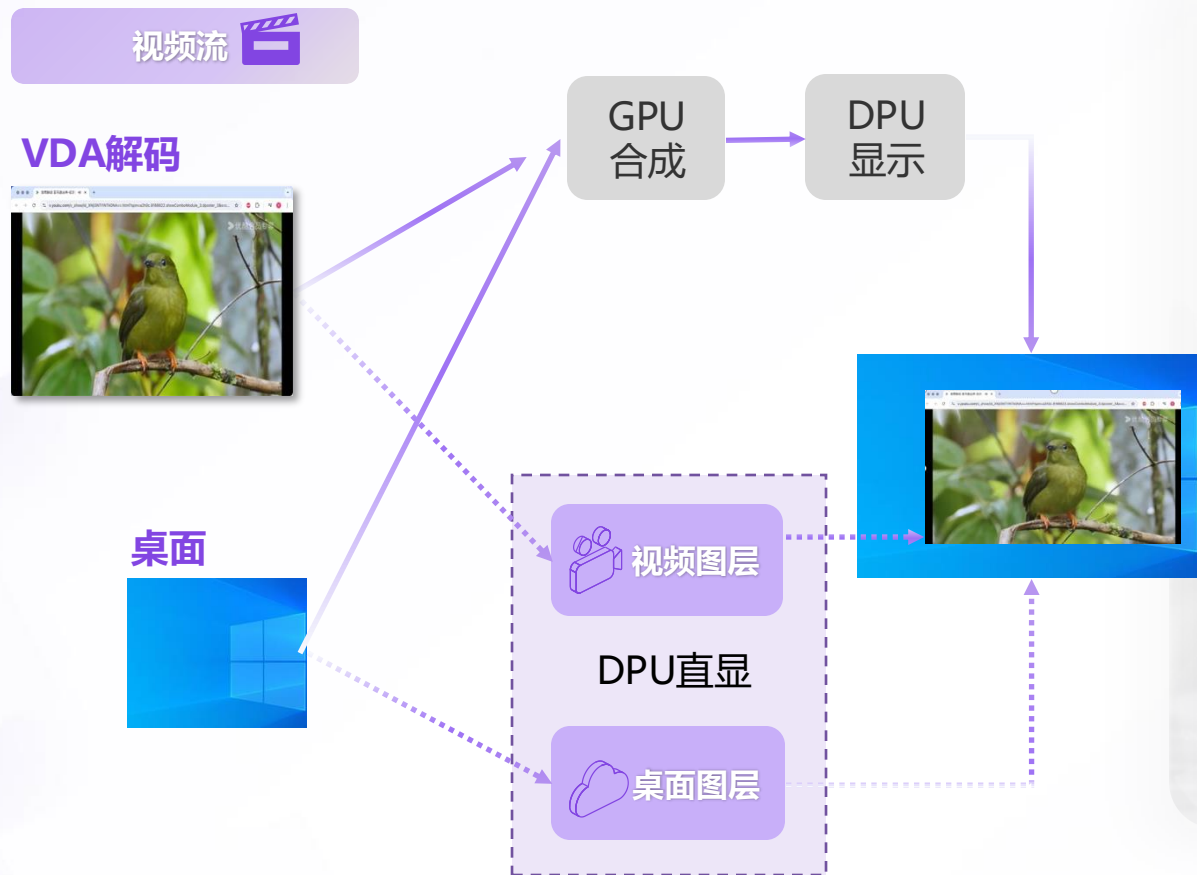


Benchmark优化提升

优化3 – 显示链路优化

VDA解码优化

- 全链路零拷贝
- 支持更丰富的硬件解码后端
- 平滑解码 4K@60fps



GPU合成

- 高GPU占用
- 显示链路Delay较大



视频直显

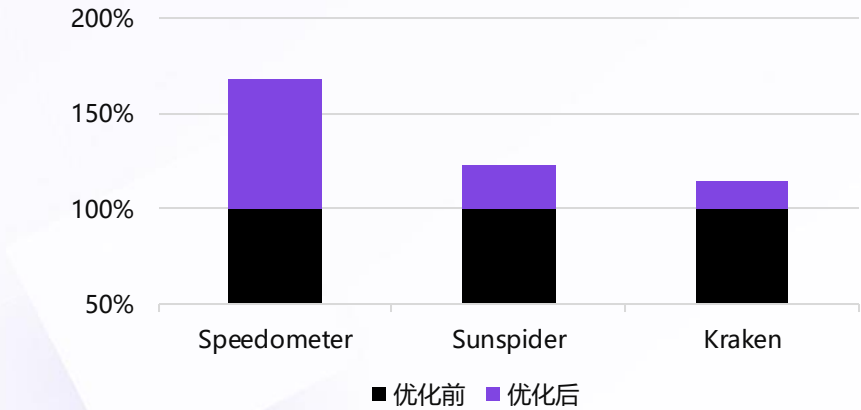
- 降低GPU占用率20~40%
- 流畅送显4K@60fps视频图像



03 | 性能优化成果

性能优化成果

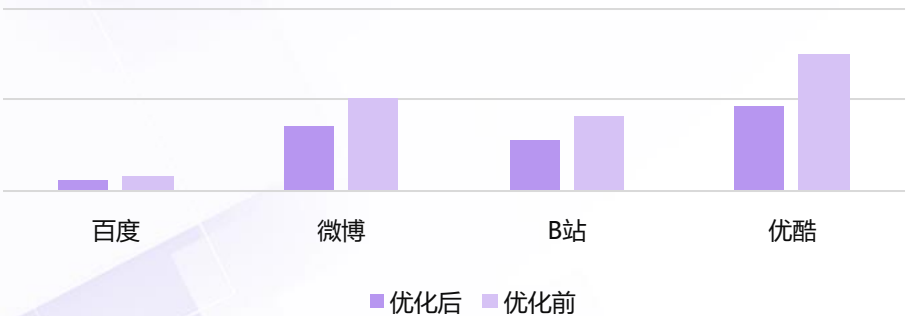
V8 Benchmark对比



视频播放

屏幕尺寸	播放视频源	优化后	优化前
1080p	1080p@60fps	流畅	流畅
	4K@60fps	流畅	卡顿
4K	1080p@60fps	流畅	非常卡顿
	4K@60fps	流畅	非常卡顿

启动时间



产品应用



Thank you



玄铁公众号



玄铁中文站



玄铁海外站

