V8 for RISC-V: What's new in 2025

智能软件研究中心
PLCT Lab V8小队 陆亚涵 邱吉
yahan@iscas.ac.cn

创新求实 · 永竞一流

报告主题

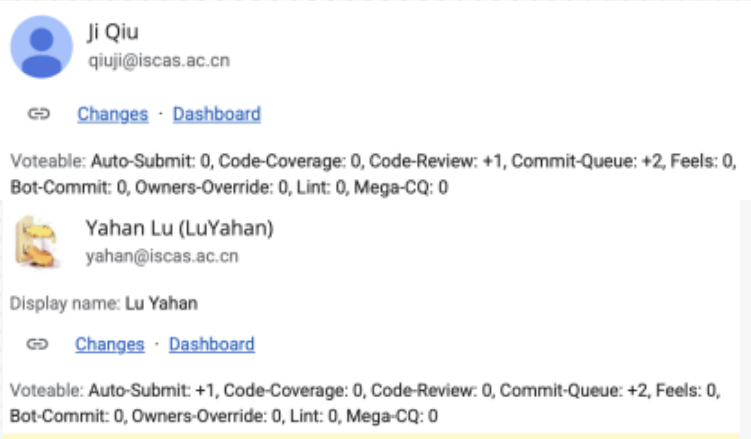
- 背景介绍
- 过去一年，V8 RISC-V所做的工作
  - 适配新的WASM/JS语言特性
  - 新增扩展支持
- 未来计划

背景介绍

- V8 是 Google 浏览器 Chrome/Node.js的 JavaScript/WebAssembly开源引擎
- Google Chrome 在当前浏览器市场中占据了绝对份额优势
- 全球超过80%的企业将Node.js作为核心服务器技术，用于构建高性能后端系统、API服务及实时数据处理平台

背景介绍

- 2020年PLCT Lab开始移植V8, 2021年V8上游成功接收了RISC-V
- 维护V8 for RISC-V超过5年，在维护V8功能性完整和可用性上做了大量工作
- 针对RISC-V,我们需要完整的后端支持，包括代码生成器/汇编器/反汇编器及嵌入式模拟器
- 合并上游以后，共提交619个commit，审阅123个commit
- 合入后，代码commit数占V8社区总数2.1%，过去一年代码commit数占V8社区总数3.77%



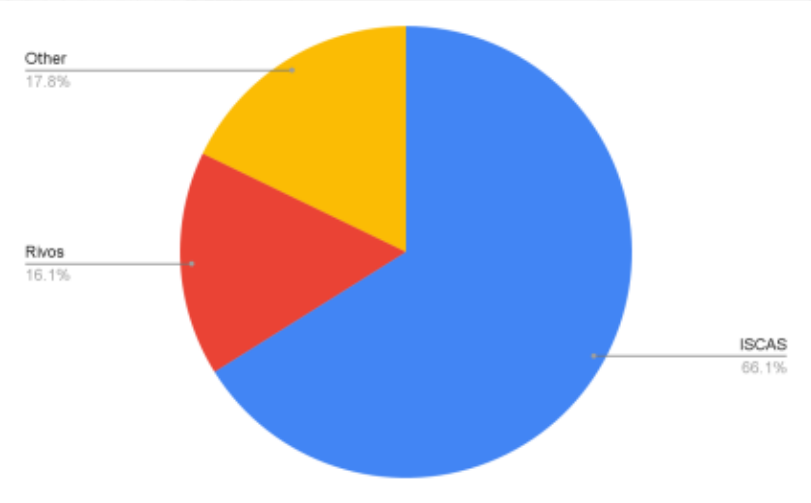
报告主题

- 背景介绍
- 过去一年，V8 RISC-V所做的工作
  - 适配新的WASM/JS语言特性
  - 新增扩展支持
- 未来计划

过去一年

- V8 RISC-V 共提交commit 219 个，新增16371行，删除13688行
- 适配新的语言特性
- 适配新的JS中间层编译器Maglev
- 适配V8 新的IR Turboshaft
- 增加了ZBA\ZBB\ZBS\Zicond扩展

V8 RISC-V贡献情况



V8 RISCV贡献比例

Thanks all the contributors, especially:
https://chromium-review.googlesource.com/q/owner:kasperl@rivosinc.com
https://chromium-review.googlesource.com/q/owner:floitsch@rivosinc.com

适配JS新语言特性

- TC39 Float16Array支持
- TC39 Base64提案支持
- BigInt 的安全性增强
- JSON 解析的增强

适配Wasm新语言特性

- Wasm RleaxedSIMD
- Wasm managed objects and garbage collection
- Wasm Fast C calls and API calls
- Fast Wasm and WebGPU interaction
- Wasm deoptimization
- Wasm JS-promise-integration/stack switching/growable stack
- Wasm memory64
- Wasm FP16
- Wasm out-of-bound trap handler

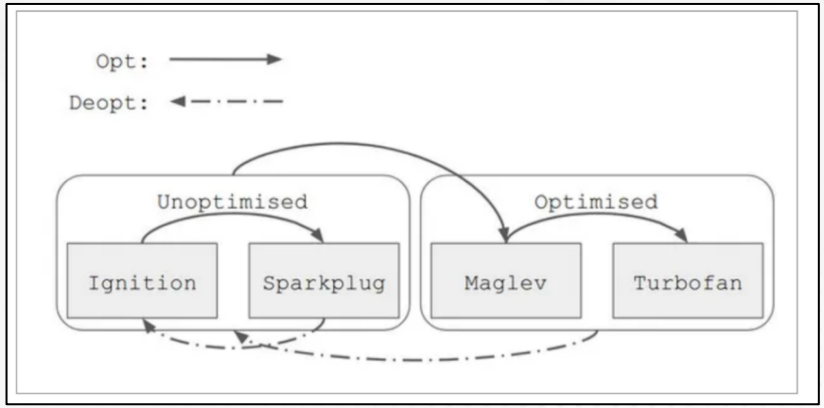
支持更多SIMD指令，移植桌面应用到浏览器内,如视频剪辑

浏览器内本地运行AI模型

支持WasmGC，可以让动态语言编译到Wasm中，支持更多软件生态

中间层JIT编译器Mgalev的支持

- 减少基线层JIT Sparkplug和顶层JIT Turbofan之间编译时间的gap
- 基于SSA和CFG
- 做一些简单的优化

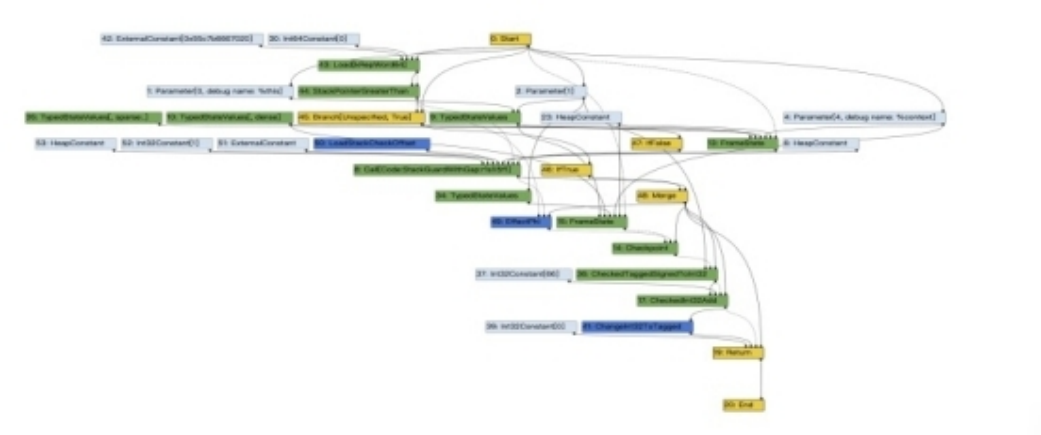


移植难点：

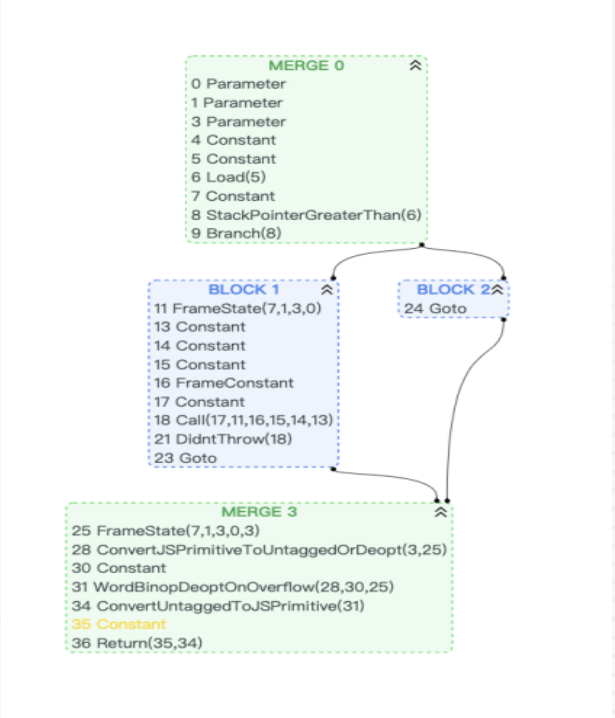
- RISC-V不是Tier-1，Maglev的设计面向Tier-1的arm/x86
- Maglev的api利用了condition寄存器，RISC-V并不具备此寄存器
- 方案：绑定一个通用临时寄存器，来作为保存condition的寄存器，从而匹配Maglev的api函数

新的基于CFG的IR Turboshaft

- 放弃了sea of node，转向Control-Flow Graph
- RISC-V目前已经全部移植完毕
- 新的IR优点：
  - 可读性更好
  - 编译分析和优化效率更高



Turbofan sea of node IR



Turbosharft CFG IR

RISCV标准扩展支持

- 支持了ZBB\ZBA\ZBS\Zicond
- 2364个Builtins函数, 622个获得了指令数目减少 （可优化比例达26%）
- Builtins函数的指令数目减少了18767条，共减少静态代码内存72KB

RISCV标准扩展支持-优化效果

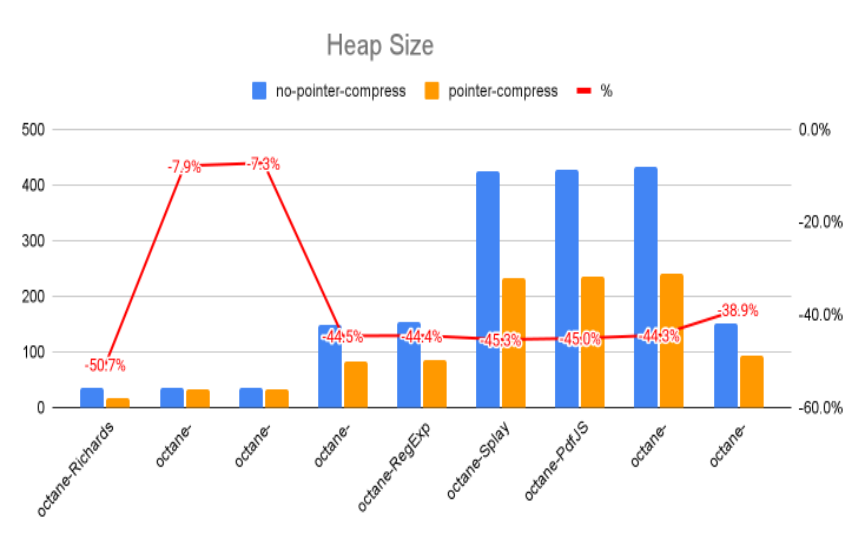
ZBB\ZBA\ZBS\Zicond

Builtins	优化前指令数	优化后指令数	变化指令数	代码尺寸减小率
MathClz32Continuation	320	160	160	100%
MathClz32	472	312	160	51.28%
WasmIntToString	3648	3264	384	11.76%
ToName	1784	1600	184	11.5%
CallForwardVarargs	164	148	16	10.81%
CallFunctionForwardVarargs	164	148	16	10.81%
NumberToString	1928	1744	184	10.55%
ConstructFunctionForwardVarargs	248	232	16	6.9%

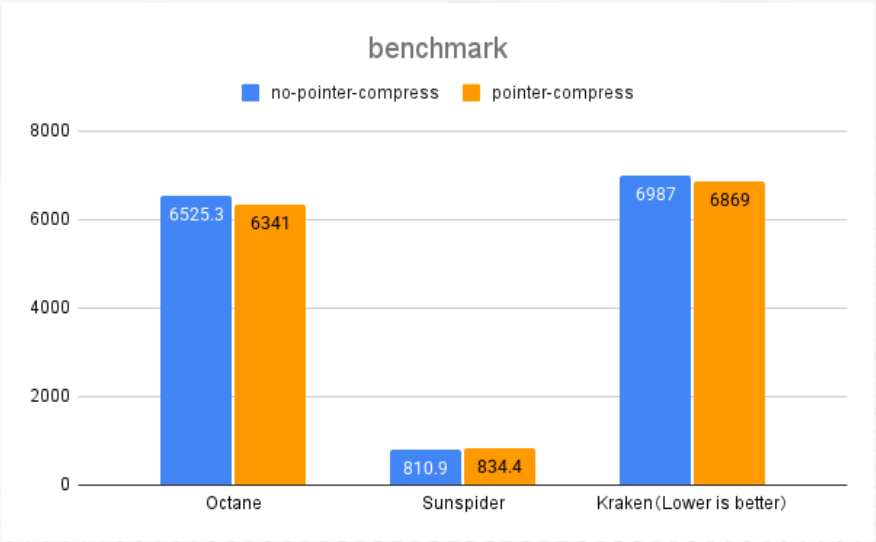
支持扩展前后，涉及的部分Builtins（内置函数）指令数目对比

内存性能数据—指针压缩优化

堆内存消耗量



指针压缩开启后性能对比



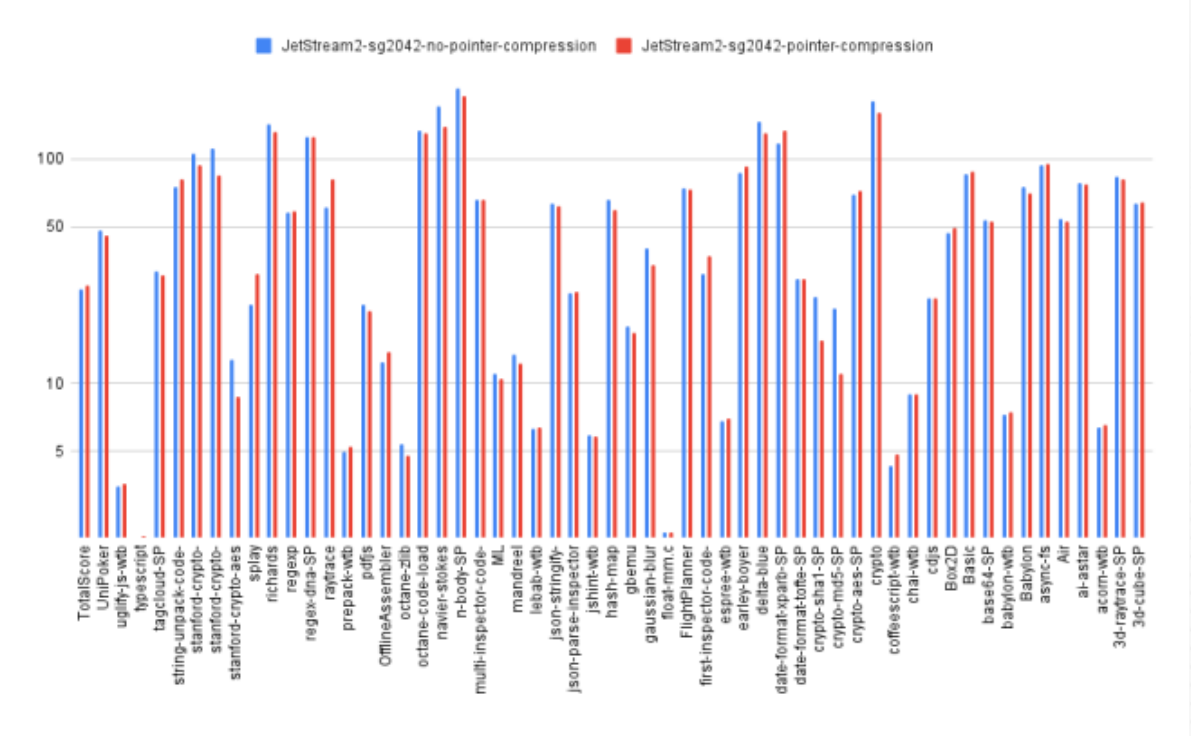
Platform: SG2042(64core,C920,2GHZ), 32GB内存

指针压缩（Pointer Compression）是V8为了优化chromium在64位平台上内存表现，将指针拆分成base（64bit）和index(32 bit),对象地址存储时只存储index.

主流JavaScript benchmark JetStream2 性能数据

Platform:
SG2042(64core,C920,2GHZ), 32GB内存

Test	Score
JetStream2-sg2042-no-pointer-compression	26.260
JetStream2-sg2042-pointer-compression	27.293



未来工作

V8高级特性实现：

- Leaptiering
- Wasm SIMD map to sizeless RVV，充分发挥RVV的性能，在未来适配更多高性能应用运行在V8 for RISC-V上

Chromium：

- 整个浏览器的系统剖析和优化，增强在高性能RISC-V平台上的用户体验

ISCAS
中国科学院软件研究所
Institute of Software Chinese Academy of Sciences

谢谢