技术铁幕下的
RISC-V 软件生态如何建立

吴伟 @ 重德智能

w@droid.ac.cn

2019-06-13

自我介绍

- 重德智能CTO
  - 提供工具链（编译器）技术服务
- HelloLLVM 和 HelloGCC 负责人
  - 工具链领域国内唯一活跃的社区，始于 2007

交流大纲

1. 什么是技术铁幕，对于国内 RISC-V 生态的影响
2. 为什么软件生态建立起来这么难
3. RISC-V 芯片公司如何胜出（一位IC门外汉的个人观点）

什么是技术铁幕

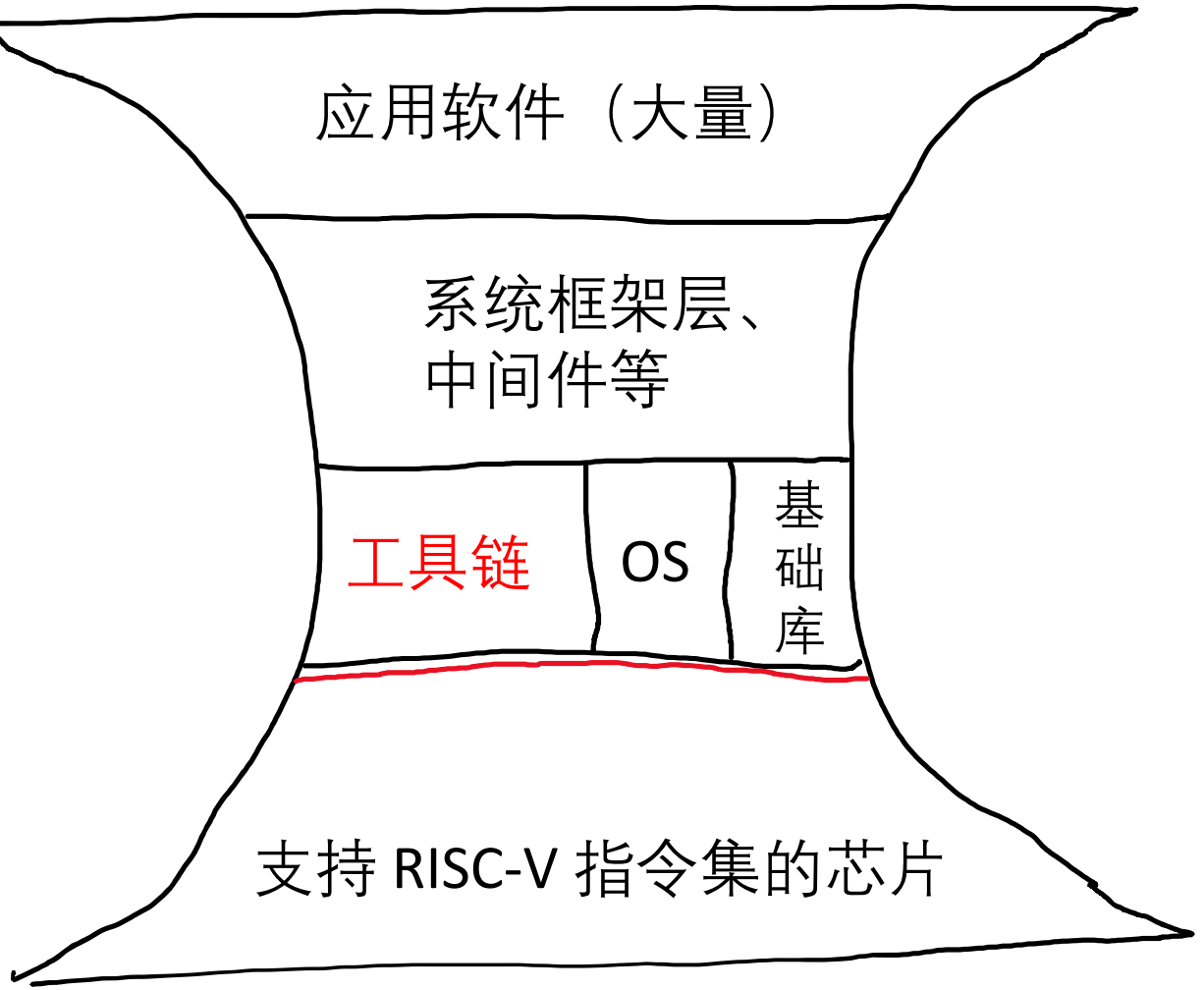
HUAWEI

对国内 RISC-V 生态：尚无实质性影响

- 开源软件本身不受技术封锁制约
- 国内外的 RISC-V 生态都处于起步阶段
  - 国内货架及在研的 RISC-V 芯片多数为 MCU 级别
- 即使没有制约，软件生态也很难建立
  - 国内尚无大型生态的成功案例

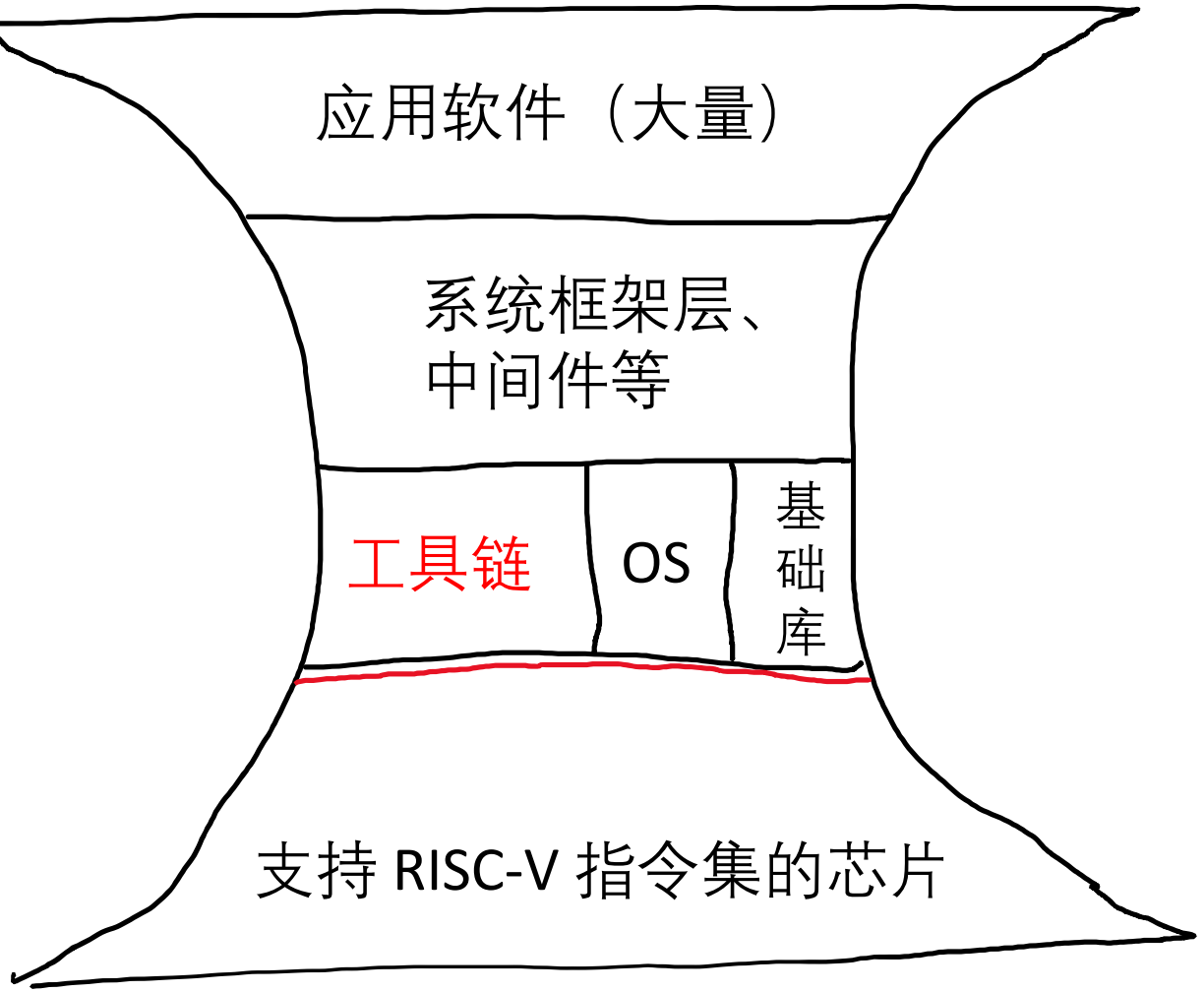
软件生态

- 用户
- 应用软件（服务）
- 系统框架层
- OS、基础库、工具链
- 芯片（CPU、MCU）



建立软件生态为什么这么难

- 用户
- 应用软件（服务）
- 系统框架层
- OS、基础库、工具链
- 芯片（CPU、MCU）



建立软件生态为什么这么难

编译路漫漫
程序语言 | 编译技术 | 开源软件

IDE | Research | Security | SpiderMonkey | Tricks | Unix/Linux | V8 | Web | 编译 | 闲谈

Home > 闲谈 > 龙芯CPU的JavaScript性能

龙芯CPU的JavaScript性能

June 14th, 2015 Go to comments Leave a comment

这是4月份的新闻的评论，一直没有时间来写，拖了两个月了。

四月份的时候，类似“龙芯性能不到iPhone6 A8性能的1/10？”的新闻[1][2][3]又一次把龙芯拉出来被人吐槽。其中出现 1/10 对比的一项是 SunSpider 测试，SPEC CINT2000 的测试差异没有那么大。另外没有贴出来 SPEC CFP2000 的分数对比。

作为一度天天跑SPEC和SunSpider等Benchmark的人来说，新闻中的测试报告是没有可信度的。性能测评需要有一套完整的流程，才能够确认自己获得的测评分数是正确的。不同的环境设定、软硬件组合、是否联网[7]等都会影响到SPEC的测评分数。

[4]中有人提供了反面的数据，表达另一种可能性。

就我个人的经验而言，龙芯的 SunSpider 性能差异更多的应该是在软件层面。现代的浏览器都通过（多个层级的）JIT来对JavaScript进行加速，开启和关闭JIT的SunSpider、Kraken、Octane跑分差异可能会相差20～30倍。使用 iPhone 的同学可以在 Safari 和微信自带的浏览器中测试对比下速度差异（iOS 中只有 Safari 可以开 JIT）。三大开源浏览器JIT引擎对于龙芯采用的MIPS后端的支持都还很初级，甚至还不能正常使用。龙芯官方[6]和开源爱好者[5]的数据也印证了龙芯

RSS

最近的日志

- 如何用TrueCrypt加密Windows系统分区
- PLDI 2016 Accepted Papers
- 技巧: 合并多份Word文档的评注信息
- 我的2015年终总结
- 大事不好, IE/Edge的JS引擎Chakra也要开源了!
- 订阅本博客的方法
- LLVM Weekly 整整100期了
- 一个小bug引发的Mozilla开发过程改进, 堪称雷厉风行
- WebAssembly 要开始并入 SpiderMonkey 了
- Firefox/SpiderMonkey的龙芯MIPS64后端

标签云

afl-fuzz Android Baseline Compiler bash benchmark C++ CCS clang Code Coverage code reading compiler Firefox Fuzz gcc Google gpu

建立软件生态为什么这么难

Firefox/SpiderMonkey的龙芯MIPS64后端

November 1st, 2015 Go to comments Leave a comment

不久之前写过一篇博客[1]为龙芯CPU"辩护", 解释为什么龙芯CPU的JavaScript性能会很低.

> 就我个人的经验而言，龙芯的 SunSpider 性能差异更多的应该是在软件层面。现代的浏览器都通过（多个层级的）JIT来对JavaScript进行加速，开启和关闭JIT的SunSpider、Kraken、Octane 跑分差异可能会相差20～30倍。使用 iPhone 的同学可以在 Safari 和微信自带的浏览器中测试对比下速度差异（iOS 中只有 Safari 可以开 JIT）。三大开源浏览器JIT引擎对于龙芯采用的MIPS后端的支持都还很初级，甚至还不能正常使用。龙芯官方[6]和开源爱好者[5]的数据也印证了龙芯CPU跑 JavaScript 的尴尬。软件层面的缺失，或者往大了说，生态环境的问题，也是龙芯（MIPS架构）需要面对的一个大问题。

但是现在情况可能就要大不同了. 中科梦兰(龙芯系)的 heiher[2] 同学在今年开始频繁地向 Mozilla 提交MIPS64后端代码[3], 相信启用了JavaScript JIT的 Firefox 在龙芯CPU会有很大的性能提升. 到时候(瞎猜)快个20倍也说不定哦 😉

[1]: 龙芯CPU的JavaScript性能

最近的日志

- 如何用TrueCrypt加密Windows系统分区
- PLDI 2016 Accepted Papers
- 技巧: 合并多份Word文档的评注信息
- 我的2015年终总结
- 大事不好, IE/Edge的JS引擎Chakra也要开源了!
- 订阅本博客的方法
- LLVM Weekly 整整100期了
- 一个小bug引发的Mozilla开发过程改进, 堪称雷厉风行
- WebAssembly 要开始并入 SpiderMonkey 了
- Firefox/SpiderMonkey的龙芯MIPS64后端

标签云

afl-fuzz Android Baseline Compiler bash benchmark C++ CCS clang Code Coverage code reading compiler Firefox Fuzz gcc Google gpu

建立软件生态为什么这么难

Firefox/SpiderMonkey的龙芯MIPS64后端

November 1st, 2015 Go to comments Leave a comment

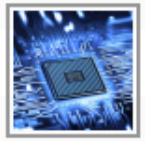
不久之前写过一篇博客[1]为龙芯CPU"辩护", 解释为什么龙芯CPU的JavaScript性能会很低.

> 就我个人的经验而言，龙芯的 SunSpider 性能差异更多的应该是在软件层面。现代的浏览器都通过（多个层级的）JIT来对JavaScript进行加速，开启和关闭JIT的SunSpider、Kraken、Octane 跑分差异可能会相差20～30倍。使用 iPhone 的同学可以在 Safari 和微信自带的浏览器中测试对比下速度差异（iOS 中只有 Safari 可以开 JIT）。三大开源浏览器JIT引擎对于龙芯采用的MIPS后端的支持都还很初级，甚至还不能正常使用。龙芯官方

最近的日志

- 如何用TrueCrypt加密Windows系统分区
- PLDI 2016 Accepted Papers
- 技巧: 合并多份Word文档的评注信息
- 我的2015年终总结
- 大事不好, IE/Edge的JS引擎Chakra也要开源了!
- 订阅本博客的方法
- LLVM Weekly 整整100期了

Leave a comment Trackback Trackbacks (0) Comments (1)



Heiher

June 22nd, 2016 at 23:01 | #1 Reply | Quote

呵呵，你预测的没错，2015年底时在龙芯3A1000上实测启用MIPS64 JIT性能提升了23倍。

建立软件生态为什么这么难

[JS-internals] Removing the MIPS back-end js-internals ×

Jan de Mooij jdemooij@mozilla.com via lists.mozilla.org Fri, Apr 12, 7:20 PM
to r, dragan.mladjenovic, yuyin-hf, JS

Hi all,

The MIPS code for SpiderMonkey's JITs has been broken for a while now and there has been some discussion about removing it. Even though it was never a Tier 1 platform, MIPS support still affects code elsewhere in the engine (things like condition codes for example) and people still spend time implementing and fixing MIPS code when making platform-specific changes. Future changes like Cranelift integration likely also require significant MIPS work.

Given all this, I'd like to propose removing the MIPS code two weeks from now. Let me know if you have any questions or concerns.

Thanks,
Jan

---

dev-tech-js-engine-internals mailing list
dev-tech-js-engine-internals@lists.mozilla.org
https://lists.mozilla.org/listinfo/dev-tech-js-engine-internals

建立软件生态为什么这么难

xwafish@gmail.com via lists.mozilla.org
to dev-tech-js-engine-internals

Apr 15, 2019, 5:00 PM

Hi,

We use Firefox extensively so we really do not want the mips port will be removed. we will fix the error soon and want maintain mips port in the future.

I come from Loongson Technology company in China, Loongson is a company which designs general processors based on mips. The market is focused on China at present. Loongson's official website: http://loongson.cn/index_en.html. The core products of loongson are general processors, so the core software such as firefox are very important for loongson.

Last years, imgtec used to maintain the MIPS port, we mainly cooperate with them to contribute code, and now we only provide firefox52 in our os, so we did not do enough work with spidermonkey inbound branch, but now I think we can do some more in spidermonkey.

thank you.

建立软件生态为什么这么难

pfgenyun@gmail.com via lists.mozilla.org
to dev-tech-js-engine-internals

Mon, Apr 15, 5:15 PM

Hi all

I am in charge of browser development in Loongson technology. In the past, we have done a lot of work based on Firefox and SpiderMonkey for our customers and contributed a lot of code in these communities. Our company is willing to invest more resources to maintain the MIPS branch, including SpiderMonkey and Firefox. My colleague YuYin（xwafish@gmail.com） and Qiao Pengchen(qiaopengchen-hf@loongson.cn) are willing to maintain MIPS branch. We hope that the MIPS branch of SpiderMonkey'Jit will not be removed from the main branch.

Thanks
fei

•••

建立软件生态为什么这么难

Kannan Vijayan kvijayan@mozilla.com via lists.mozilla.org
to pfgenyun, JS

Apr 16, 2019, 12:09 AM

There are significant constraints that are being imposed on other parts of the codebase to support this extra architecture branch. From the perspective of a production spidermonkey build, it often lags behind in feature support, correctness, and bugfixes. It imposes API requirements on the masm, and requires work whenever there are cross-system patches (e.g. changing the boxing format as in: https://bugzilla.mozilla.org/show_bug.cgi?id=1401624).

With the current design, every additional CPU architecture imposes a cost on all the other implementations (mostly due to API changes needed, and convoluted expression of concepts shared across all of them). The ideal way to move forward with MIPS would be to shift development towards a general code-generator backend that can support multiple architectures.

To the longsoon stakeholders: would there be openness to considering contributing a MIPS codegen implementation to CraneLift (https://github.com/CraneStation/cranelift)? The team is interesting to eventually transitioning to a single codegen backend. It's not on the immediate roadmap due to priority, but we intend on moving in that direction as opportunity allows. If we can find some common path on the roadmap that allows us to reduce the independent maintenance burden of the architecture, we can manage support for this architecture much better.

Regards.

建立软件生态为什么这么难

pfgenyun@gmail.com via lists.mozilla.org
to dev-tech-js-engine-internals

Apr 16, 2019, 5:10 PM

In the past, the mips port of many open source project including SpiderMonkey and firefox ware maintained by imgtec team. They had done a lot of work and done well. We mainly cooperated with them to do it. In the following work, we will gradually assume the responsibility of the open source community（mips port）including SpiderMonkey and firefox.The bugs will be fixed soon.
Cranelift is a very meaningful project and we will arrange engineers to submit to the mips port as soon as possible. Maybe mipe64el port at first.

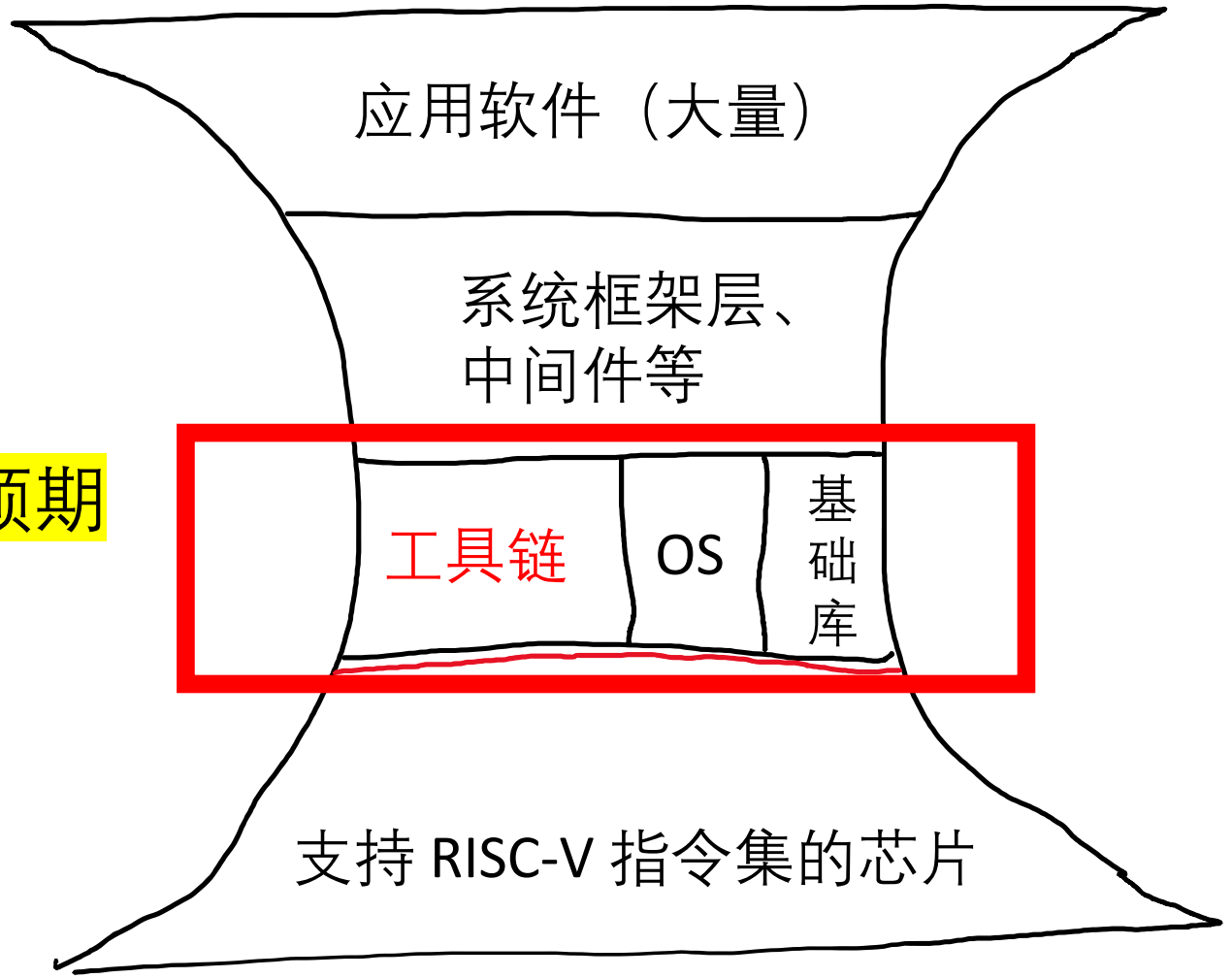
Thanks
fei

在 2019年4月16日星期二 UTC+8上午12:09:30，Kannan Vijayan写道：

•••

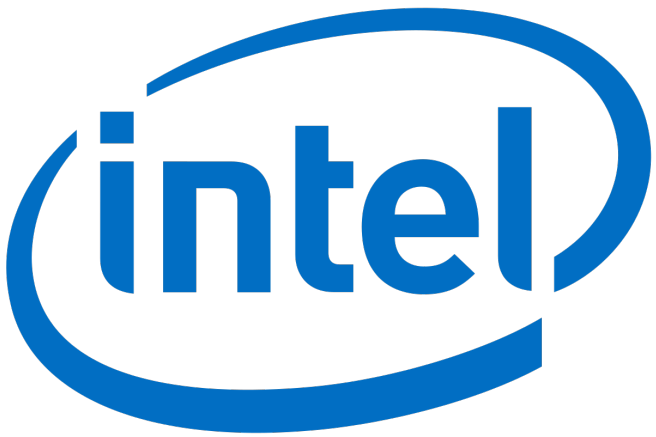
建立软件生态为什么这么难

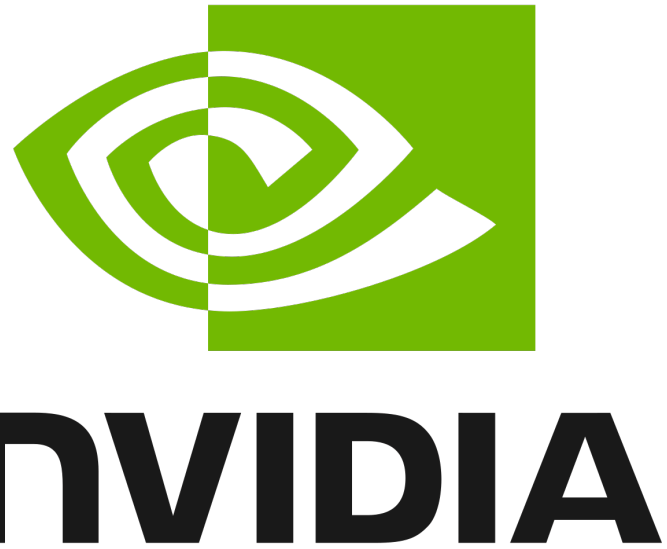
- 前期需要大量投入
  - 高级别技术研发人员
  - 所需资金远多于管理层预期
  - 需要不断修复和积累
  - 需要维护和建立社区



RISC-V 新兴公司如何胜出

- 观察：所有市场统治地位的芯片公司均有庞大的生态





借助 RISC-V 生态：二档起步

- RISC-V 基金会及社区建立了良好的基础环境
- 短期内对标ARM建立软件生态支持
- RISC-V 芯片公司可以基于现有生态快速起步

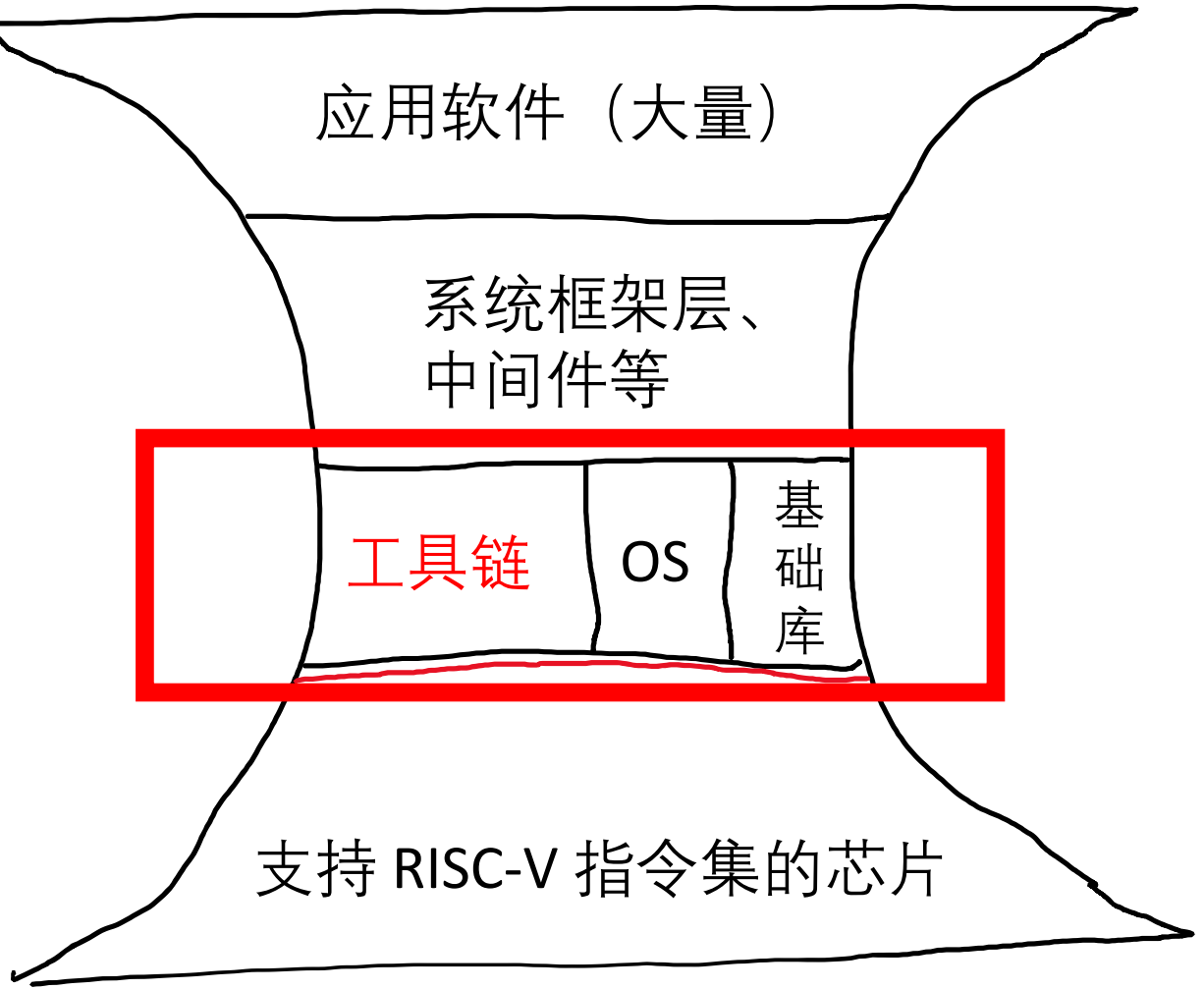
后续提速，没有顺风车

- 同样ISA，不同微架构

  性能和能耗可以有巨大差异

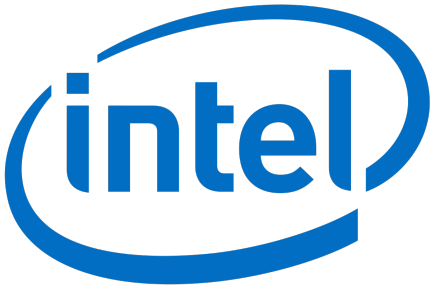
- 扩展指令集需要自行支持

- 整体性能优化需要自己做



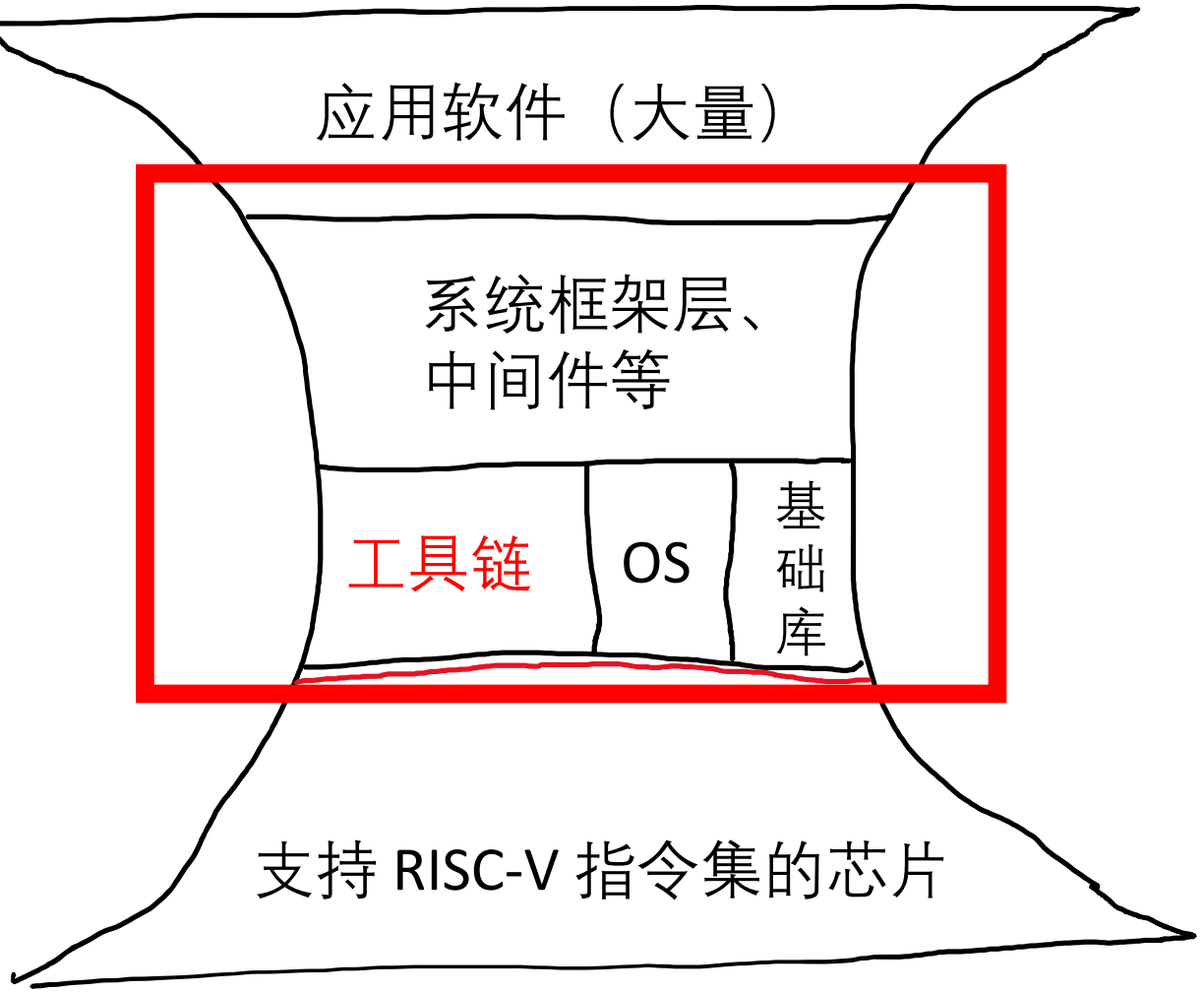
没有顺风车

- 真实段子：Intel 编译器黑 AMD 性能跑分
- 开源编译器无视兆芯X86，稍加改进之后40%以上提升



从龙芯等前辈身上学到的经验教训

- 重视软件生态
- 重视软件生态
- 重视软件生态



感谢各位，欢迎提问

也欢迎添加微信后续技术交流（我负责技术部分）

BACKUP：DISCUSSIONS

- NVIDIA最核心的技术是什么？最害怕的对手是谁？
- 为什么工具链领域招资深开发很难？
- 编译器开发的难度在什么地方？

技术铁幕下的
RISC-V 软件生态如何建立

吴伟 @ 重德智能

w@droid.ac.cn

2019-06-13