

程序员

被浏览

358,485

编程经历中有没有曾经遇到以为一个很简单的问题，最后折腾很久才解决？

他们也关注了该问题

关注问题 写回答 邀请回答 1 条评论 分享 邀请回答 举报

查看全部 137 个回答

Felis sapiens

函数式编程、编程语言、编程 话题的优秀回答者

草莓大福、RednaxelaFX、Belleve、Glavo、霜月琉璃等 96 人赞同了该回答

刚接触 Haskell 时，写 Lisp 类语言的解释器遇到了问题：如何在一门 lazy 的语言里表达 strict 的语义。解释器架构很 naive，就是在 Maybe monad 里做错误处理，还老是在各种地方该规约时不规约，折腾下来只好用到处插 seq 和 bang patterns 强迫求值的方法勉强解决，还要写一些“单元测试”，通过是否陷入死循环（超过预设 timeout 就算）判断有没有 lazy 语义泄露过来，比如  $(\lambda x y \rightarrow x) (\lambda x \rightarrow x) ((\lambda x \rightarrow x x) (\lambda x \rightarrow x x))$

当时为了这问题差些劝退 Haskell 转 OCaml，两相对比，在 lazy 语言中写 strict 的解释器不如在 strict 语言中写 lazy 的解释器容易。后来和 @阅千人而惜知己 面基时提到这事他哈哈大笑，说这是个典型问题，应该对解释器做 CPS 变换，分离 host 语言和 target 语言的求值策略，他还给我发了个讲义，回去照着一试果然成功。后来读了 John Reynolds 大名鼎鼎的《Definitional Interpreters for Higher-Order Programming Languages》才真正悟到其中道理。

编辑于 2017-05-11

赞同 96 14 条评论 分享 收藏 感谢

朱元

cort\_proto作者(dwz.cn/7v6Lin)请指教

192 人赞同了该回答

用C++写一个有栈协程，协程的栈设置为4096时，运行效率比4096-8或者4096+8慢了40%左右。。。简直快要把脑袋钻到显示屏里了。。

结果努力的读了又读N-way associative cache相关的内容，最终确定仅仅是CPU cache的问题。。还好2天之内就定位到问题了。

这么多人赞就分享一个相关的图（和我的协程无关，但情况完全一致，测试用例设法避免了别的影响，让4K的慢显得特别明显。）



关于作者

Felis sapiens

函数式编程、编程语言、编程 话题的优秀回答者

电影旅行敲代码、Antokha Yuuki、暮无井见铃也关注了她

回答 234 文章 40 关注者 14,871

已关注 发私信

相关问题

编程的世界是什么样的？ 47 个回答

如何高效编程？ 16 个回答

哪些大神是一只手编程的？ 5 个回答

函数式响应型编程(Functional Reactive Programming)会在什么问题上有优势？ 12 个回答

自学成才的程序员是在哪学到那些知识的？ 19 个回答

相关推荐



陈硕  
编程、

出错了！请稍后再试。

137 人赞同了该回答

在测试 SSL handshake 性能的时候，发现用单线程性能显著高于两个线程。单线程是在client和server之间切换角色，而两个线程分别是 client 和 server。按说除了一点同步的开销，性能应该一样（加密计算在 client / server 之间交替进行，没有重叠），实际相差很大。后来发现我的台式机 CPU 自动降频，单线程是单核全速运行，而两个线程是两个核半速运行，当然慢了。解决办法：  
[How I can disable CPU frequency scaling and set the system to performance?](#)

[展开阅读全文](#) ▼

[查看全部 137 个回答](#)



森懂物理学：理解世界的简指南

共 31 节课

▶ 试听



Live 面试你

了!

★★★★★ 1298 人参与



挑战编程技能：57 道程序员功力测试题

965 人读过

▶ 阅读



[刘看山](#) · [知乎指南](#) · [知乎协议](#) · [隐私政策](#)

[应用](#) · [工作](#) · [申请开通知乎机构号](#)

[侵权举报](#) · [网上有害信息举报专区](#)

[违法和不良信息举报](#)：010-82716601

[儿童色情信息举报专区](#)

[电信与服务业务经营许可证](#)

[网络文化经营许可证](#)

[联系我们](#) © 2018 知乎

