



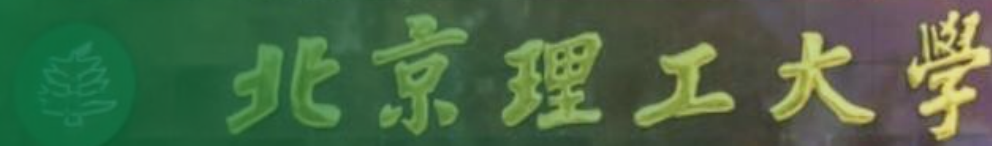
ReL4操作系统中 基于用户态中断的异步系统调用设计

◎ 答辩人：陈伟豪

◎ 导师：陆慧梅

2024年5月31日

德以明理 学以精工



论文终稿查重

提交至校内查重平台进行查重，结果为1.3%。

盲审结果

第一位专家评分：A（82分） 无修改意见

第二位专家评分：B（78分） 无修改意见

盲审均分：A（80分）

论文评优

通过形式审查，参与评优答辩



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

目录 / CONTENTS

01

研究背景

02

ReL4微内核介绍

03

设计与实现

04

实验与结果分析

05

总结与展望

PART ONE

研究背景



- 由于微内核结构的特殊性，低IPC与系统调用成本至关重要
- 相关工作
 - 提升多核利用率
 - 采用Notification机制增强同步IPC等
 - 减少内核路径
 - 虚拟消息寄存器、快速路径（Fast Path）等
 - 减少特权级切换开销
 - 组合系统调用、ASID机制等



- 前两部分优化趋于极限
- 减少特权级切换开销已成为优化主要方向^[1]
- 对于频繁调用的场景，用户态-内核态的上下文切换已经成为限制系统性能的最大瓶颈



■ 基于用户态中断^[2]的改进方案

- 用户态中断通过硬件的方式，无需陷入内核即可发送信号
- 避免了特权级切换，更好的通信性能

■ [1] Gu J, Wu X, Li W, et al. Harmonizing Performance and Isolation in Microkernels with Efficient Intra-kernel Isolation and Communication[J]. 2020.

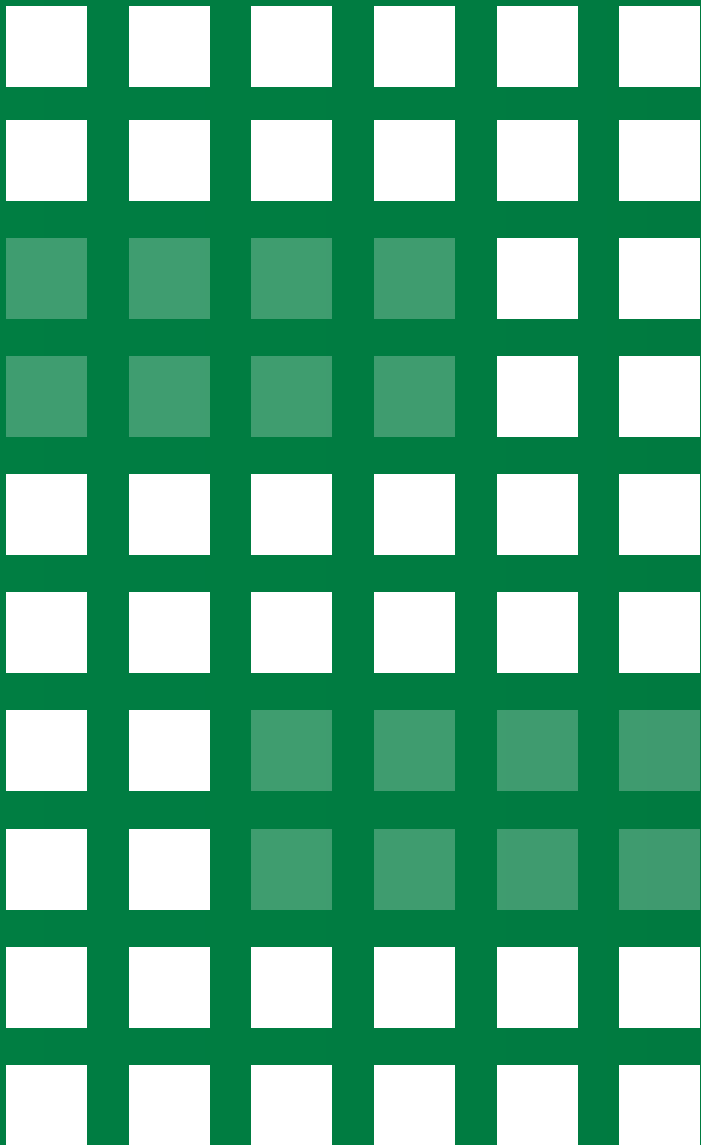
■ [2] Sohil M. User Interrupt-A faster way to signal[R]. LPC Virtual, Linux Plumbers Conference. 2021.9.

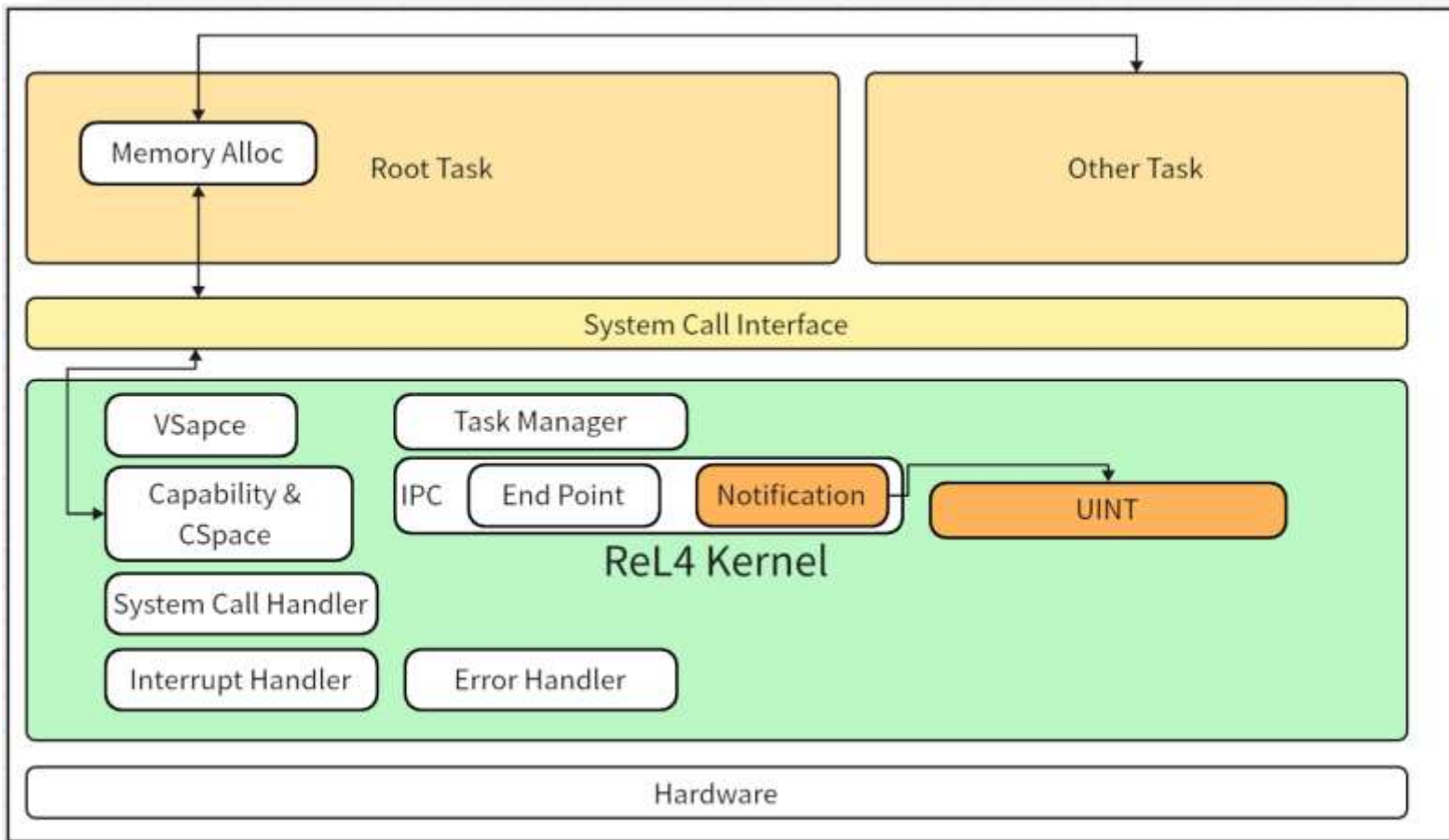


- 利用用户态中断改进后的Notification机制
- 异步化改造同步系统调用
- 减少特权级切换频率，从而降低系统调用开销，提升系统性能

PART TWO

ReL4微内核介绍

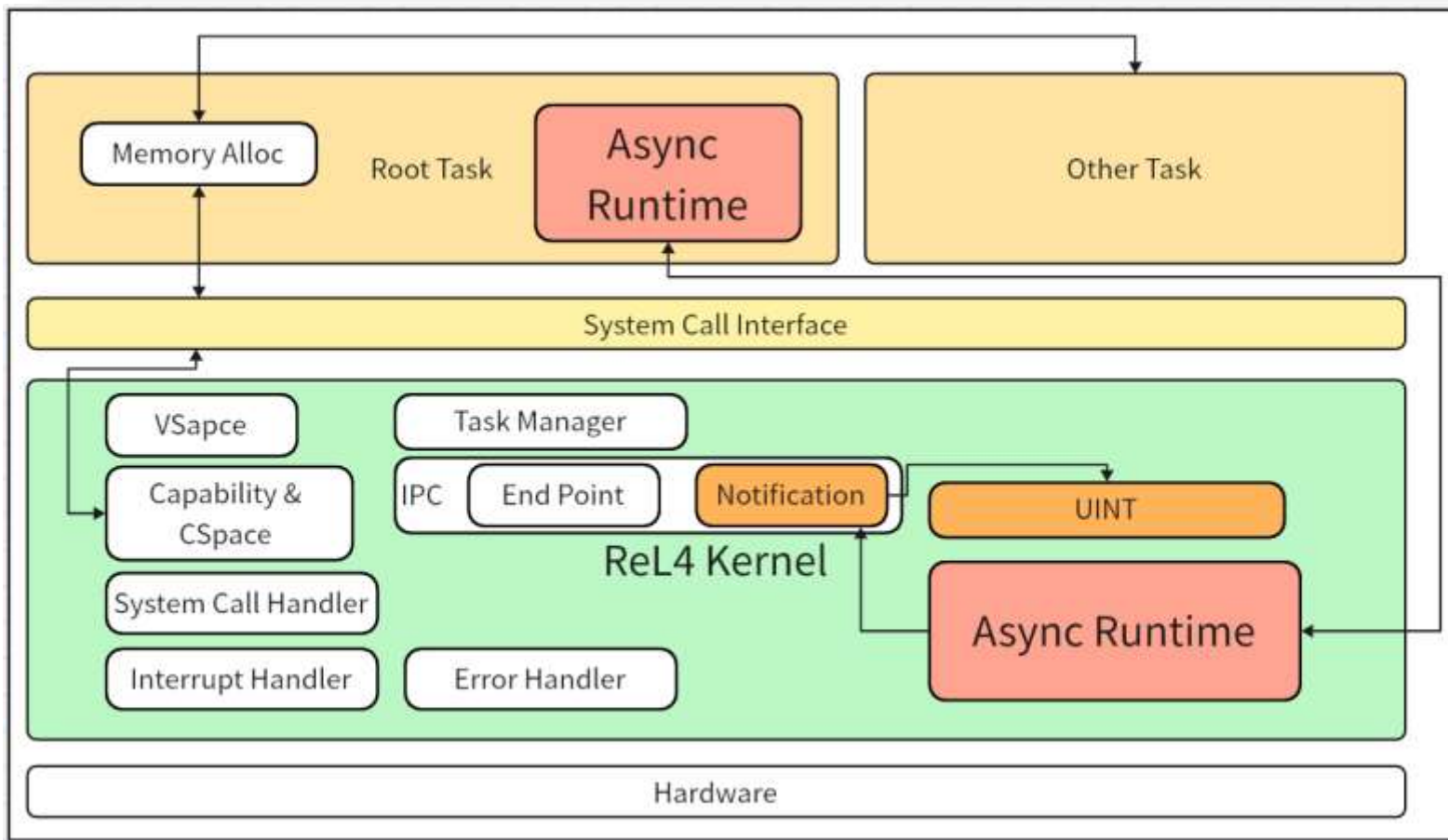




- 作为开发平台
- 用Rust重写seL4^[3]支持RISC-V接口的微内核^[4]
- 提供基本的内核支持功能

图1 ReL4架构图

- [3] Klein G, Elphinstone K, Heiser G, et al. seL4: Formal verification of an OS kernel[C]//Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles. 2009: 207-220.
- [4] ReL4微内核的开发者为北京理工大学2022级硕士研究生廖东海学长



- Async Runtime模块:
- 实现了异步系统调用的相关方法
- 以基于用户态中断（UINT^[5]）的 Notification机制为基础

图1 ReL4架构图

■ [5] 田凯夫. RISC-V 用户态中断扩展设计与实现[D]. 北京: 清华大学, 2023.

PART THREE

设计与实现

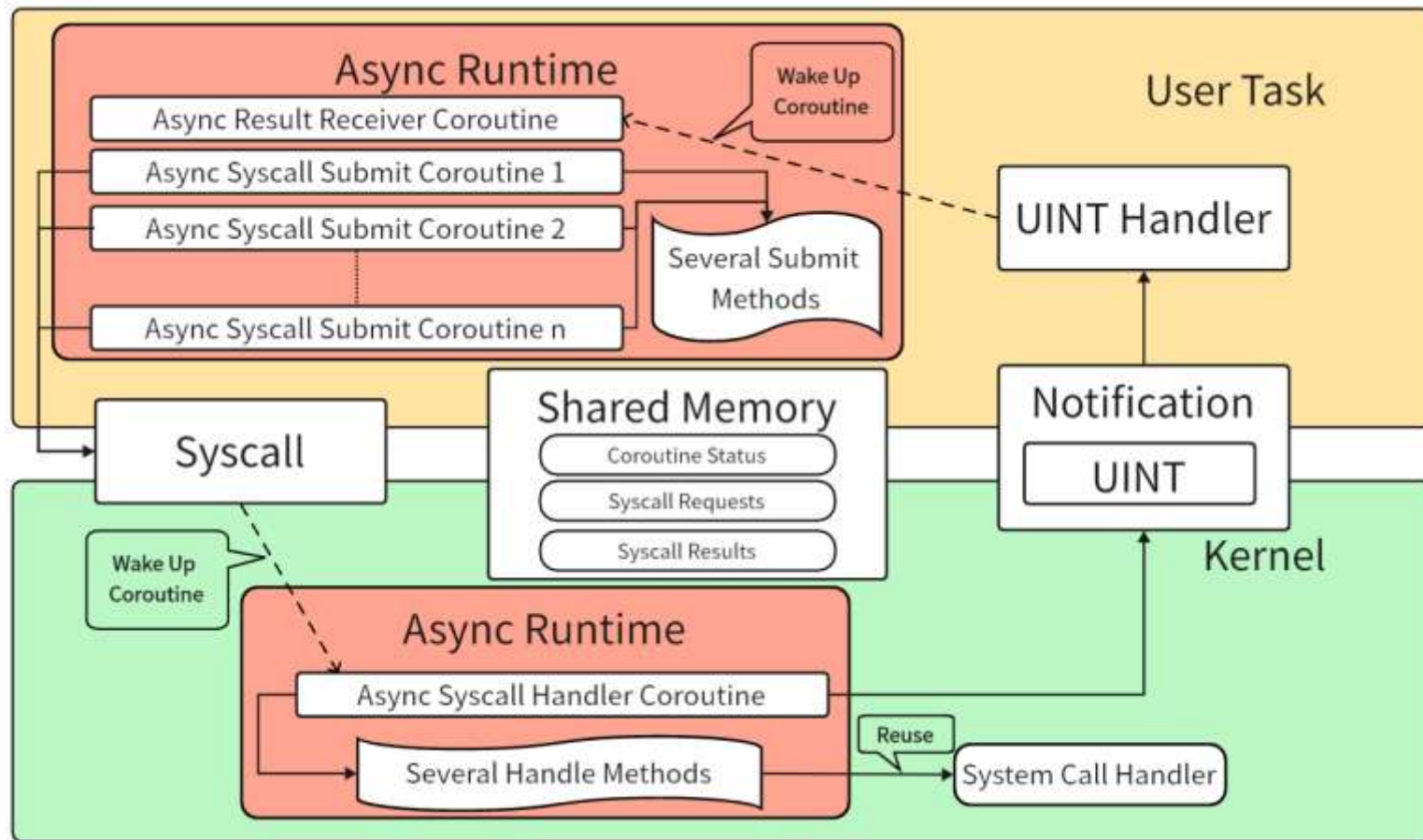
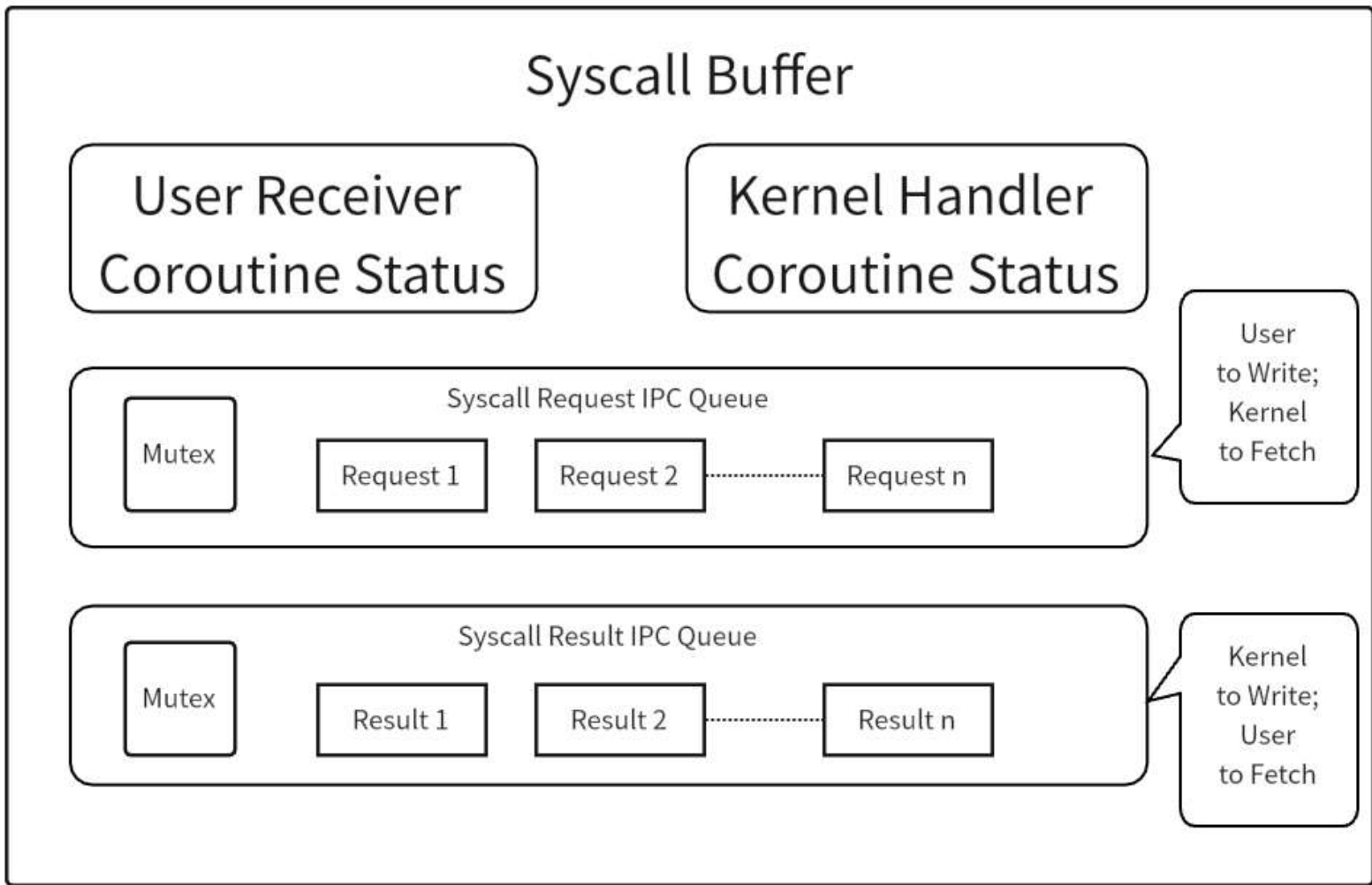


图2 异步系统调用框架图

- 用户态、内核态都配备异步运行时模块
- 用户态异步运行时:
 - 1个接收协程+n个提交协程
 - 提交方法
- 内核态异步运行时:
 - 1个处理协程（任意核心运行）
 - 处理方法
- 用户态-内核态交互:
 - 通过协程
 - Syscall、共享内存、Notification



- 问题1：共享内存实现与互斥访问控制
- 问题2：系统调用区分与参数传递
- 问题3：协程实现



- 对应问题1
- 协程状态标识
- IPC Queue储存请求和响应
- IPC Queue中元素为IPC Item
- 解决共享内存并发访问:
 - 协程状态标识采用布尔原子
 - IPC Queue使用互斥锁Mutex

图3 共享内存实现图



■ 对应问题3

■ 对应问题2

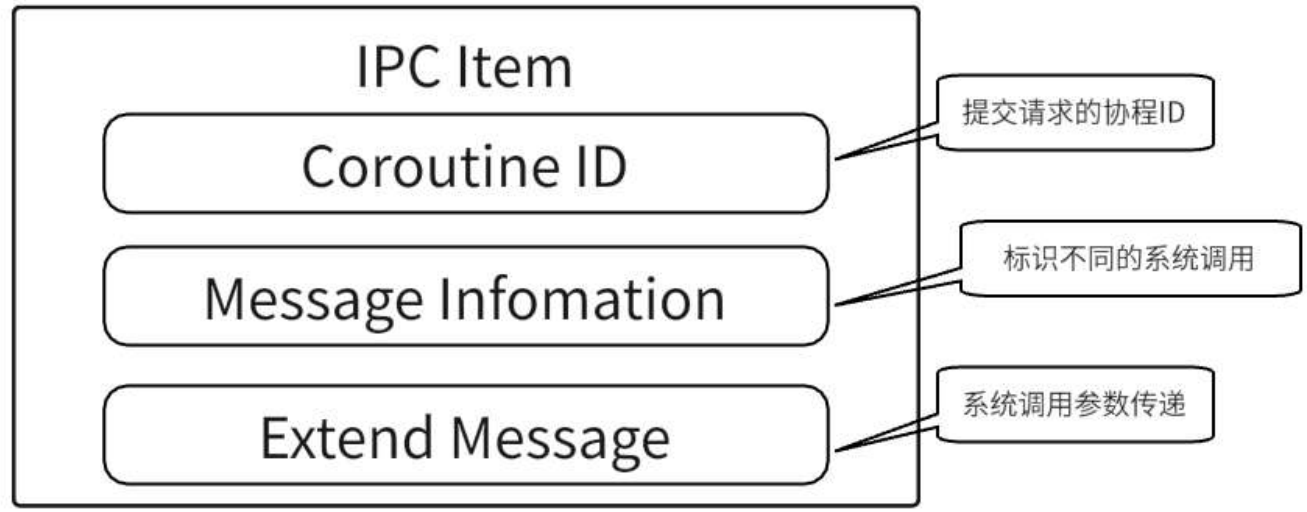


图4 IPC Item实现图

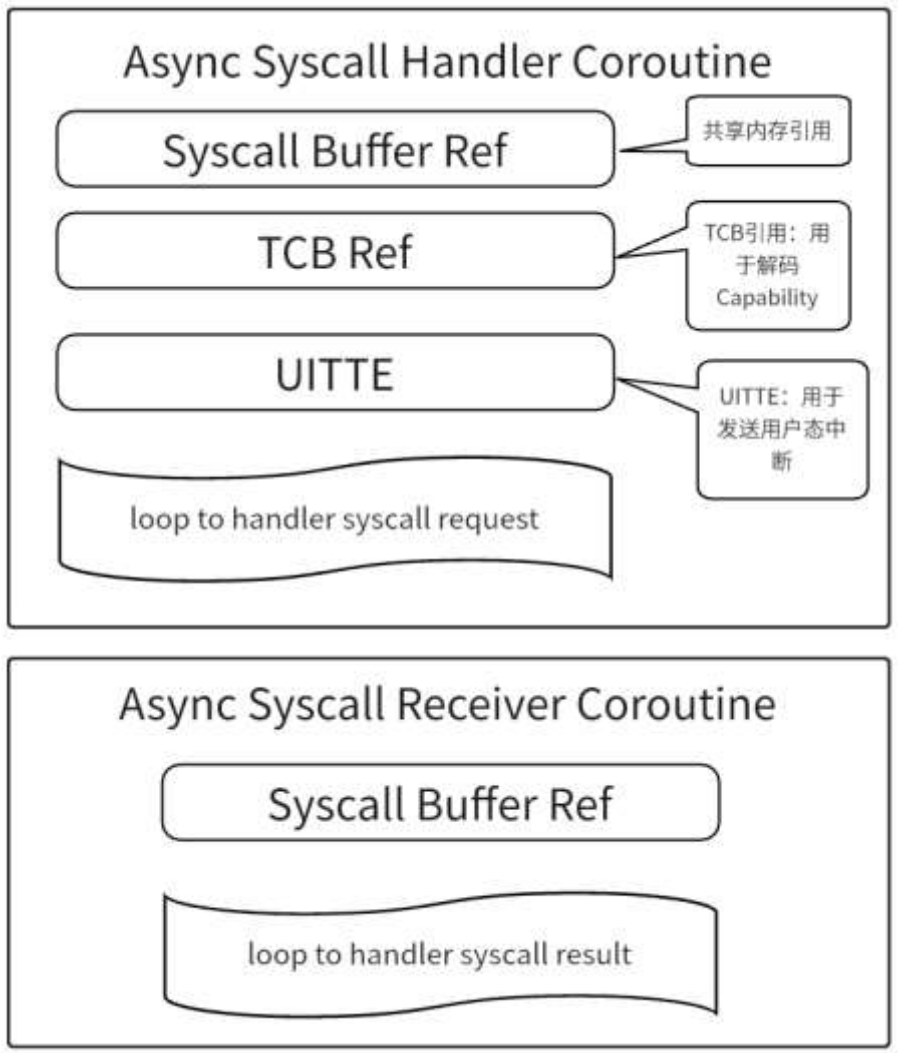


图5 协程实现图

PART FOUR

实验与结果分析



- 输出类系统调用可以如预期输出内容。
- Notification机制类系统调用可以实现Notification内核对象与TCB内核对象的绑定与解绑，在绑定与解绑操作前后的对象的物理地址变化符合预期。
- 内存映射类系统调用可以实现虚拟地址到物理地址的映射，映射后可以通过虚拟地址访问内存，解除映射后再访问内存会出现段错误。
- 通用类系统调用在上述测试中能够正确配合使用，符合预期。



- 实验场景：内存映射分配
- 实验参数：不同并发度（提交协程数量）
- 对比对象：异步 VS 同步
- 评价指标：每轮耗时（处理速度）、陷入频率（特权级切换开销）
- 运行环境：FPGA开发板（Qemu模拟器版在论文中）

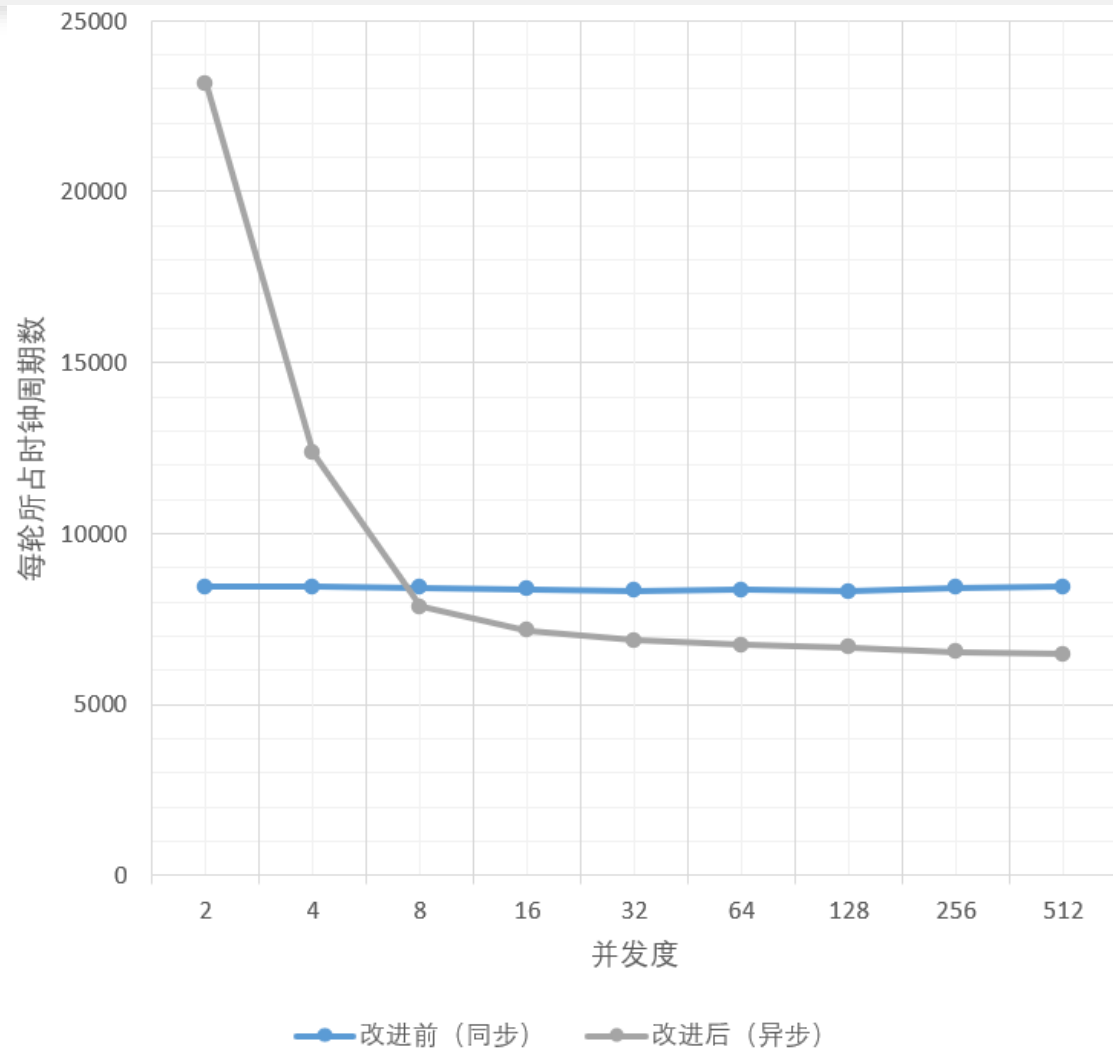
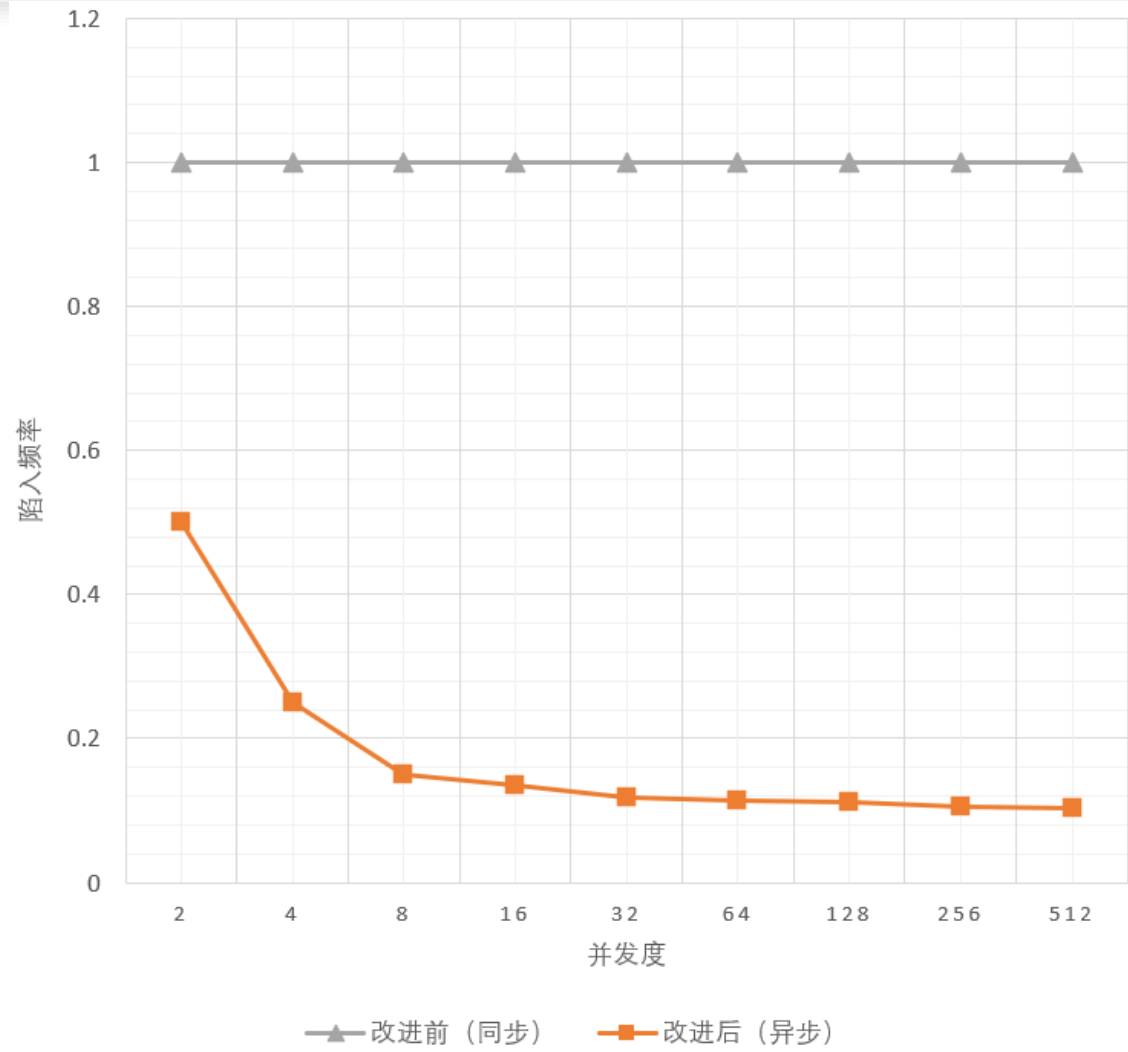


图6 基于每轮耗时指标的性能测试图

- 每轮所占时钟周期数表征处理速度
- 并发度表征并发协程数
- 低并发环境下:
 - 改进后性能低于改进前行性能
 - 原因：内核负载低，内核协程经常阻塞
- 高并发环境下:
 - 改进后性能超过改进前性能
 - 改进后性能达到改进前实现的130.81%
 - 原因：内核负载提升，内核协程稳定运行



- 陷入频率表征特权级切换开销
- 改进前的陷入频率恒为1
- 改进后陷入频率远低于改进前
- 达到了减少特权级切换频率的目的

图7 基于陷入频率指标的性能测试图

PART FIVE

总结与展望



- 所做工作：
 - 创新点：
 - 异步化
 - 结合用户态中断与共享内存
 - 并发、集中处理大量系统调用
 - 实现异步系统调用框架
 - 结果表明在高并发环境下异步系统调用具有优于原有实现的性能
- 未来发展：
 - 优化异步系统调用框架性能、增加内核协程数量、实现调度器

感谢各位专家老师 请您批评指正

◎ 答辩人：陈伟豪

◎ 导师：陆慧梅

北京理工大学

BEIJING INSTITUTE
OF TECHNOLOGY

德以明理 学以精工