Hypocaust, a RISC-V Type-1 Hypervisor

天津大学 齐呈祥
<kuangjux@outlook.com>

Overview

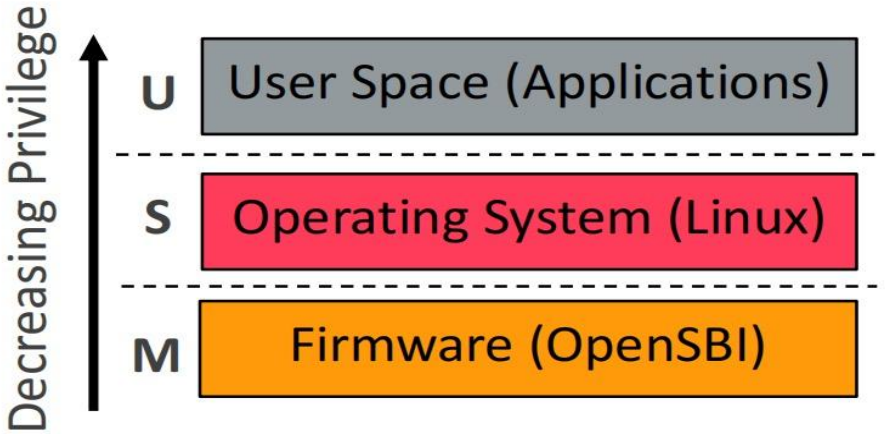
- RISC-V H Extension Overview
- Hypocaust/Hypocaust-2 Overview
- Hypocaust-2 Design & Implement
- Status & Future Work
- Questions

RISC-V H Extension

The RISC-V Hypervisor Extension

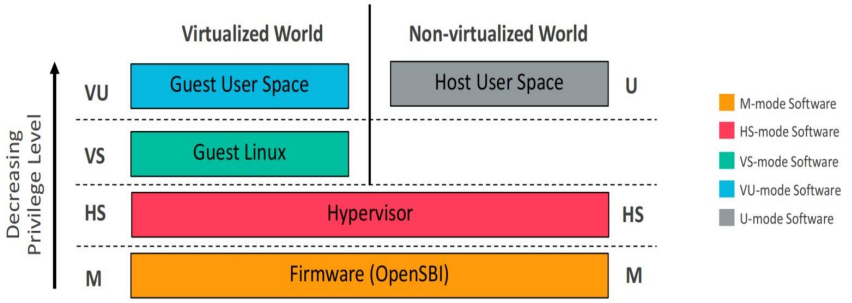
Classical RISC-V Privilege

- 32 个通用寄存器
- 3 种特权级模式：Machine(M mode), Supervisor(S mode), User(U mode)
- 每种特权级模式下都有控制状态寄存器(CSRs)
- XLen(machine word length) 可能是 32 bit，64 bit 或 128 bit

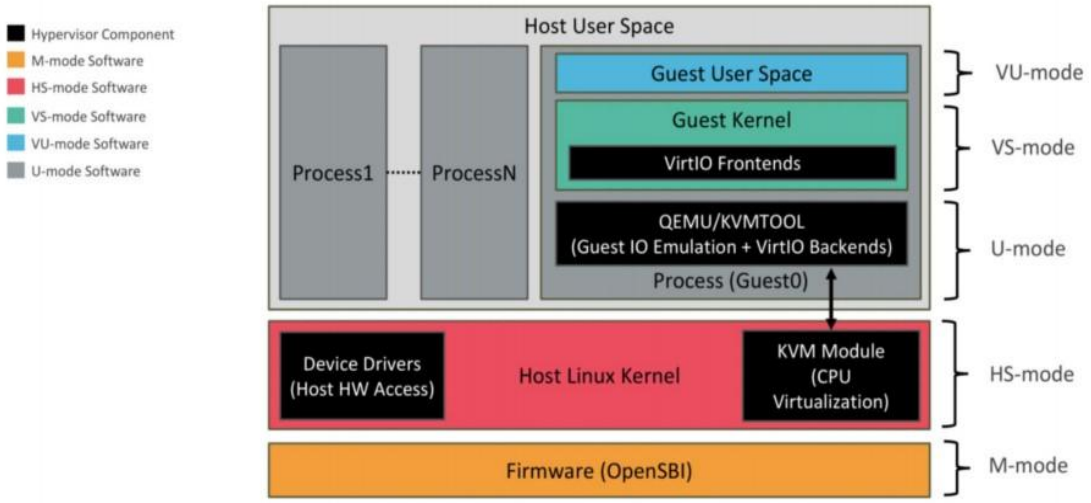


RISC-V CPU Virtualization

- HS-mode： S mode with hypervisor capabilities and new CSRs
- VS-mode: Virtualized S-mode, 用于代替传统 S-mode
- VU-mode: Virtualized U-mode, 用于代替原有的 U-mode



- 模式切换：
  - ecall： 由 VS-mode 进入 HS-mode
  - sret： 由 HS-mode 进入 VS-mode



RISC-V HS-mode CSRs

- HS-mode:
  - “s<xyz>” 指向真正的 “s<xyz>”
  - “hs<xvy>” 用来提供虚拟化功能
  - “vs<xvy>” 用来指向 VS-mode 中的虚拟 “s<xyz>” 寄存器
- VS-mode:
  - “s<xyz>” 指向 “vs<xyz>”

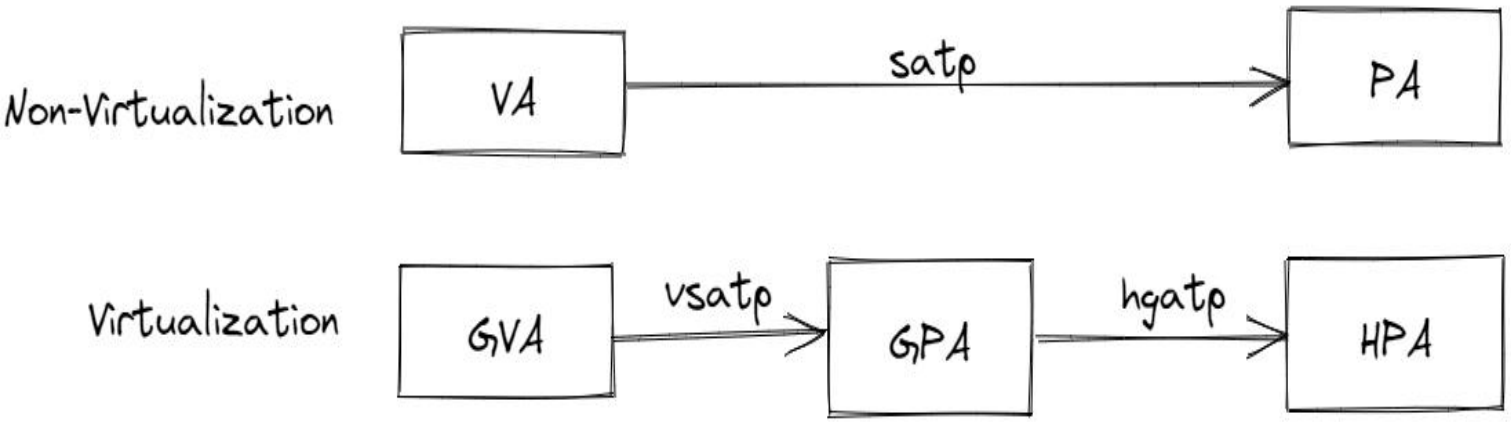
HS-mode CSRs for hypervisor capabilities	
hstatus	Hypervisor Status
hideleg	Hypervisor Interrupt Delegate
hedeleg	Hypervisor Trap/Exception Delegate
htimedelta	Hypervisor Guest Time Delta
hgatp	Hypervisor Guest Address Translation

HS-mode CSRs for accessing Guest/VM state	
vsstatus	Guest/VM Status
vsie	Guest/VM Interrupt Enable
vsip	Guest/VM Interrupt Pending
vstvec	Guest/VM Trap Handler Base
vsepc	Guest/VM Trap Progam Counter
vscause	Guest/VM Trap Cause
vstval	Guest/VM Trap Value
vsatp	Guest/VM Address Translation
vsscratch	Guest/VM Scratch

RISC-V Memory Virtualization

- vsatp: Virtual Supervisor Address Translation and Protection Register
- hgatp: Hypervisor Guest Address Translation and Protection Register

- 两阶段翻译：GVA ----> GPA(vsatp) ----> HPA(hgatp)



RISC-V Interrupt Virtualization

- PLIC：
  - 不支持 MSI
  - 不支持中断投递
  - 需要在 hypervisor 模拟 PLIC 并进行中断注入

- AIA(Advanced Interrupt Architecture):
  - IMSIC(Incoming Message Signaled Comtroller): 支持 MSI（消息信号中断），支持 IPI Virtualization
  - VS-mode下运行的客户操作系统对设备中断（作为 MSI）的直接控制，减少了虚拟机监视器（hypervisor）的干预
  - APLIC：可以更高效地处理中断

RISC-V Device Virtualization

- 设备模拟：
  - 纯软件模拟，甚至可以模拟不存在的设备
  - 平台稳定，不需要特殊的硬件支持
  - 性能低
- 设备直通：
  - VM 独占 Guest
  - 性能高，实现简单
  - 需要大量设备（假设有 100 个 VM）
  - 解决方案：
    - 恒等映射：
      - 无需实现 IOMMU，可以通过配置内存来实现 DMA 分配
      - 需要单独为每个平台配置内存
    - IOMMU：
      - RISC-V IOMMU 草案，安全性高，可移植性好，实现较为复杂

Hypocaust/Hypocaust-2 Overview

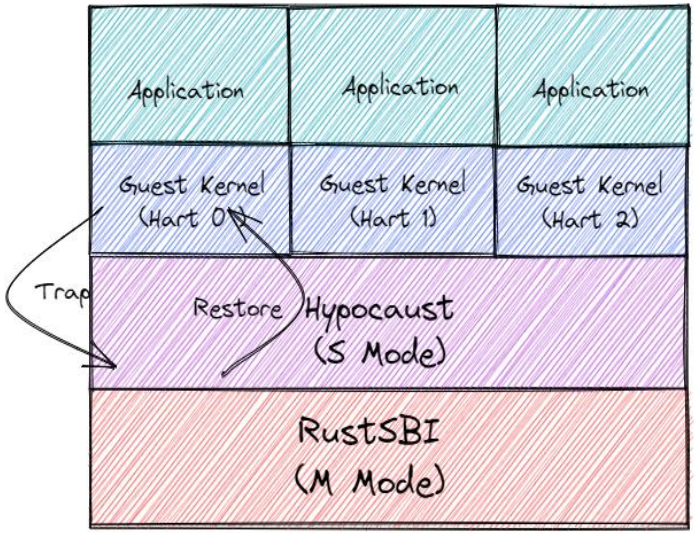
Hypocaust/Hypocaust-2 Overview

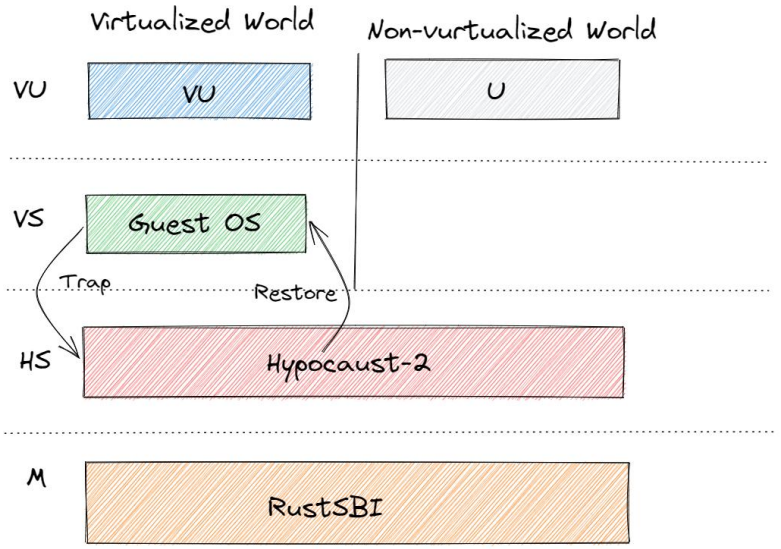
Hypocaust:

- S-mode 陷入 & 模拟
- S-mode Shadow CSRs 模拟
- Shadow Page Table
- 中断/异常注入
- 设备透传
- ...

Hypocaust-2:

- 基于设备树配置
- RISC-V H 扩展辅助虚拟化
- 两阶段页表翻译
- 异常代理
- 中断注入(PLIC 模拟)
- 设备透传
- ...





Hypocaust

第一版使用 Rust 编写的 Type-1 RISC-V hypervisor，使用 S mode trap & emulate 技术

优点：

- 可以在任何 RISC-V 平台运行，无需支持 H extension

缺点：

- 需要单独维护 vCPU CSR 的 Shadow State
- 维护 Shadow Page Table 和 Guest Page Table 的同步关系较为复杂
- 每次读写 CSRs/SFENCE.VMA 都需要进行陷入处理，性能低
- 需要为每个 VM 的每个进程都维护一个 Shadow Page Table，浪费内存

Hypocaust-2

第二版使用 Rust 编写的 Type-1 RISC-V hypervisor，使用 RISC-V H extension 提供的硬件辅助虚拟化技术

优点：

- 使用硬件辅助虚拟化技术，不需要单独为 vCPU 维护状态
- 使用 H extension 提供的两阶段页表翻译功能，不需要维护 Shadow Page Table
- 可以选择代理 VS-mode 的中断/异常，不需要每次都陷入并处理/转发中断、异常
- 实现更简单，性能更强

缺点：

- 只能在实现了 H extension 的平台上使用

Hypocaust-2 Design & Implement

Device tree based configuration

- 两种类型设备树：
  - Host Device Tree:
    - hypocaust-2 启动时配置，用于初始化内存以及外设
    - 由 SBI 提供
  - Guest Device Tree：
    - 由 hypocaust-2 在初始化 guest 的时候使用，用于构建 guest 页表以及初始化 guest 设备
    - 需要预先放在某个地址或者从文件系统中加载

Hypocaust-2 CPU Virtualization

- 硬件实现了 H extension，CPU 自动为 S-mode 维护 vs-<xyz> 寄存器，当 guest 写入 s-<xyz> 的时候自动写入 vs-<xyz> 寄存器
- 当发生 vmentry/vmexit 切换时对某些必要的寄存器进行保存
- 1-1 vCPU : pCPU 映射，无需 hypervisor 进行 guest 调度，更为简单高效

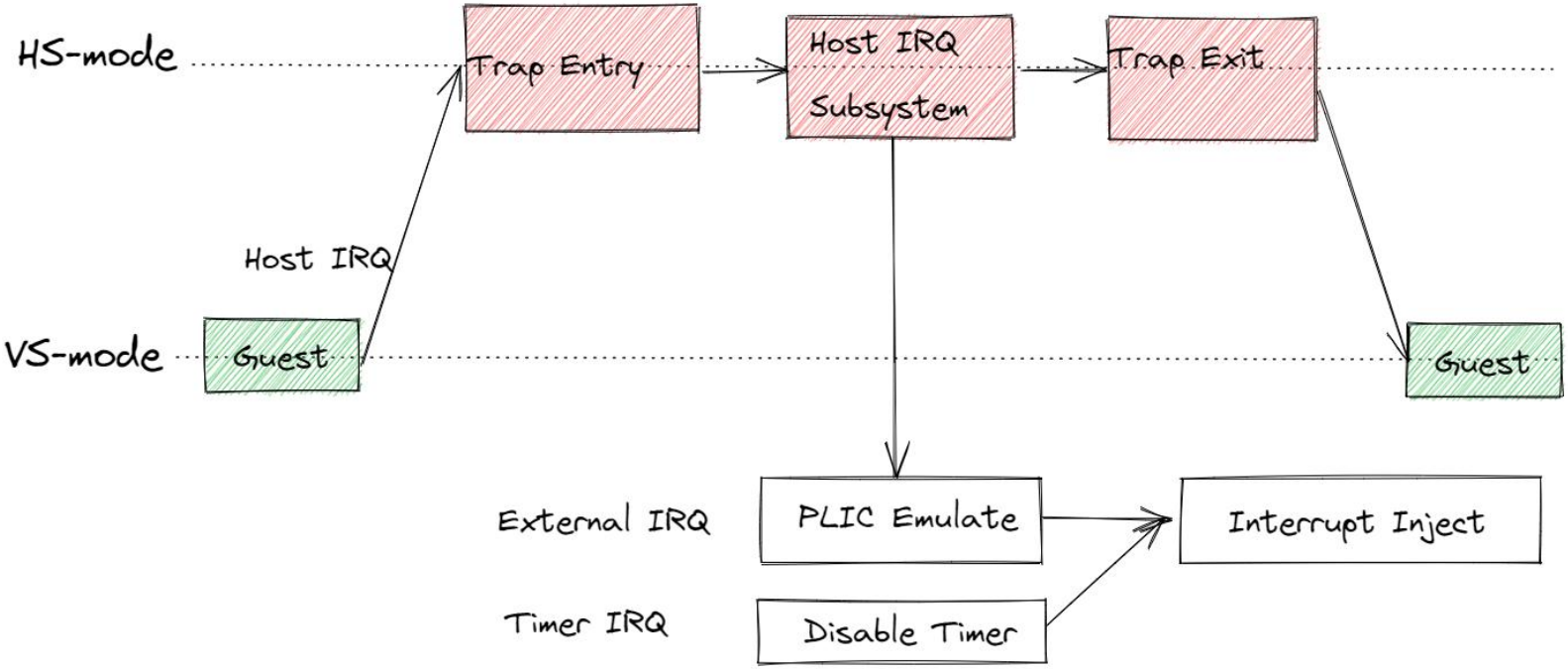
Field	Description	Save/Restore
zero	Zero register	---
ra	Return address register	Trap Entry/Exit
sp	Stack pointer register	Trap Entry/Exit
gp	Global pointer register	Trap Entry/Exit
tp	Thread pointer register	Trap Entry/Exit
a0-a7	Function argument registers	Trap Entry/Exit
t0-t6	Caller saved registers	Trap Entry/Exit
s0-s11	Callee saved register	Trap Entry/Exit
sepc	Program counter	Trap Entry/Exit
sstatus	Shadow SSTATUS CSR	Trap Entry/Exit
hstatus	Shadow HSTATUS CSR	Trap Entry/Exit
sp_exec	Stack pointer for traps	Trap Entry/Exit

Hypocaust-2 Memory Virtualization

- 为 Host 和 Guest 分别维护不同的 MemorySet 以隔离
- hypocaust-2 根据设备树配置为 Guest 分配内存并进行页表映射，根据 RISC-V H Extension，根页表需要 16KiB 对齐并且大小应该为 16KiB
- 将第二阶段页表的标志位全部设置为 RWXU，维护起来较为简单但安全性不高，最终需要修改

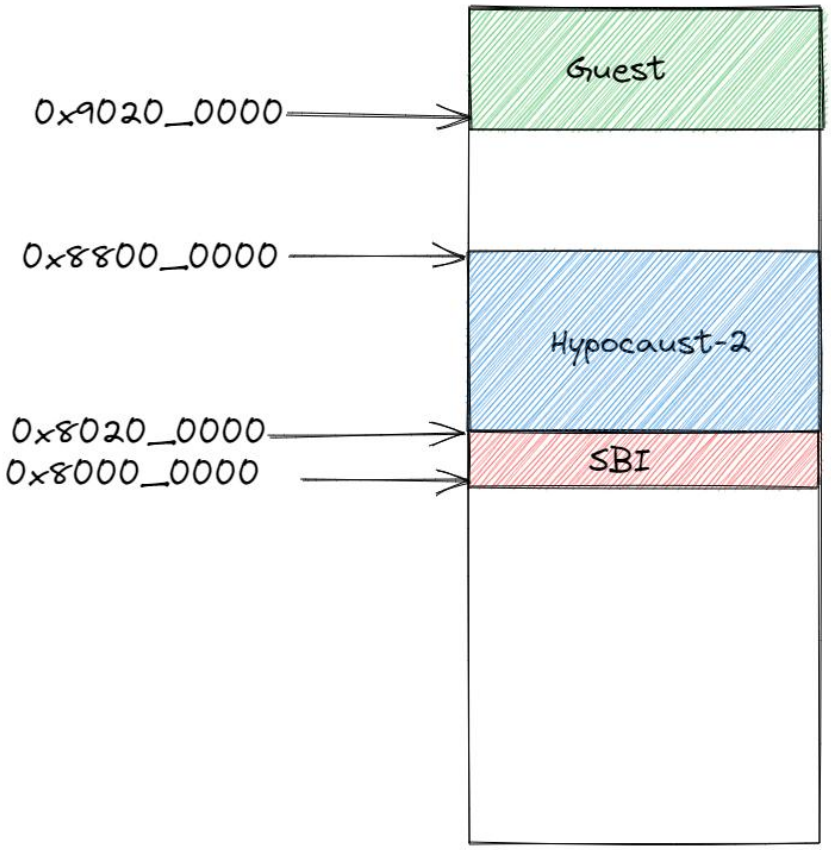
Hypocaust-2 Interrupt Virtualization

- 未实现 AIA，无法将中断代理到 VS-mode，需要进行中断注入
- 时钟中断：接收中断 -> 关闭中断 -> 中断注入 -> 收到 SBI Call -> 打开中断
- 外部中断：接收中断 -> 模拟 PLIC -> 中断注入 IRQ ID -> 接收 Guest 写 complete 寄存器 -> 告知 PLIC 中断完成



Hypocaust-2 Device Virtualization

- 使用 PassThrough 实现设备虚拟化
- 未实现 IOMMU，需要根据设备树内存配置，将 Guest 放在高地址进行恒等映射，从而使用 DMA allocation



Status & Future Work

Current Satus & Future Works

- 2023 年 2 月开始开发，当前：
  - 仅仅可以在 QEMU >= 7.0.0 版本中启动
  - 可以启动 rCore-Tutorual-v3，RT-Thread，Linux mainline
  - 使用 Rust 编写，抽象度高，代码量小，易于维护和扩展

- 展望：
  - 扩展为多核，多 guest
  - 为 hypocaust-2 实现 IOMMU/AIA
  - 等待支持 RISC-V H extension 的芯片并移植
  - 将 hypocaust-2 做成 Linux Kernel Module（兼容 KVM API）

Questions