ESWIN

RISC-V Hypervisor Extension

ESWIN 中央研究院 软件中心

基础开发部

Shawn Liu

2021-08-26

目录

ESWIN

RISC-V H-Extension Status

what needs to be virtualized

RISC-V Nested Virtualization

RISC-V KVM

RISC-V Virtualization Goals

1. Virtualized S-mode to support running guest OS under Type-1, Type-2 and hybrid hypervisors.

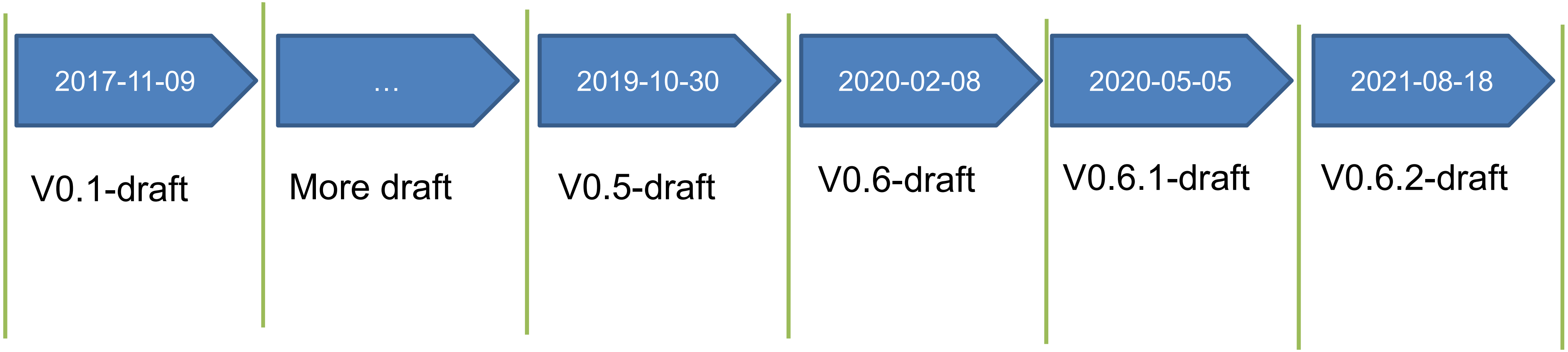
2. Support recursive virtualization

3. Be performant and parsimonious

RISC-V H-Extension Status

ESWIN

RISC-V H-Extension draft release history:



2017-11-09 V0.1-draft

… More draft

2019-10-30 V0.5-draft

2020-02-08 V0.6-draft

2020-05-05 V0.6.1-draft

2021-08-18 V0.6.2-draft

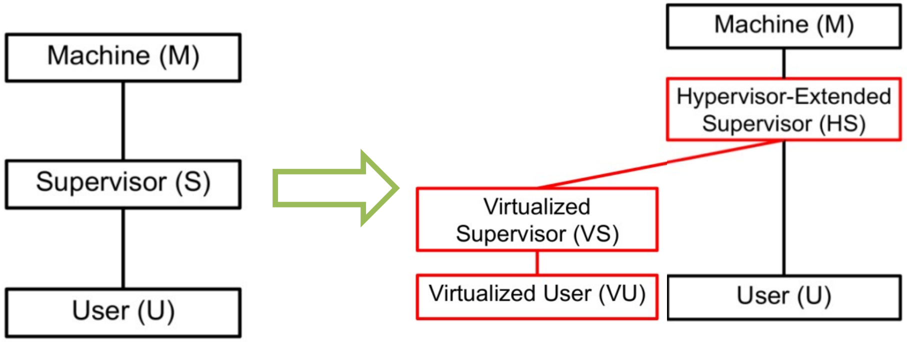
The hypervisor specific ISA in RISC-V is called RISC-V H-Extension

What Needs to be Virtualized?

ESWIN

Privilege:
The hypervisor extension changes supervisor mode into hypervisor-extended supervisor mode (HS-mode, or hypervisor mode for short)

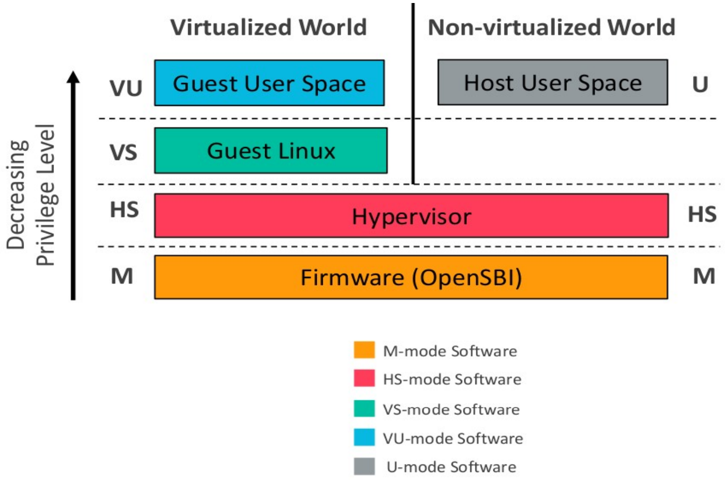
CSR:
An OS or hypervisor running in HS-mode uses the supervisor CSRs to interact with the exception,interrupt, and address-translation subsystems. Additional CSRs are provided to HS-mode.



Memory virtualization:
The hypervisor extension also adds another stage of address translation, from guest physical addresses to supervisor physical addresses, to virtualize the memory and memory-mapped I/O subsystems for a guest operating system.

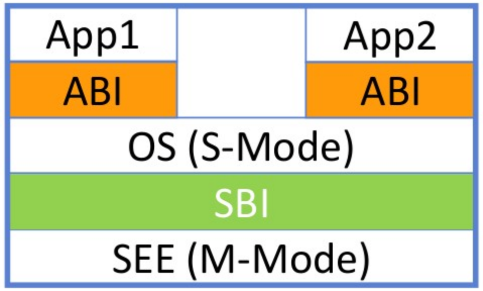
RISC-V H-Extension: Privilege Mode Changes

- □ Three privilege modes:
  - ➢ Machine (M-mode)
  - ➢ Supervisor (S-mode)
  - ➢ User (U-mode)
- □ Supported combinations:
  - ➢ M (simple embedded systems)
  - ➢ M, U (embedded systems w/protection)
  - ➢ M, S, U (Unix systems)
- □ S-mode gains new features,becomes HS-mode
- □ Two additional modes
  - ➢ Virtualized Supervisor (VS)
  - ➢ Virtualized User (VU)



Virtualized World
Non-virtualized World
Decreasing Privilege Level
VU Guest User Space
Host User Space U
VS Guest Linux
HS Hypervisor HS
M Firmware (OpenSBI) M

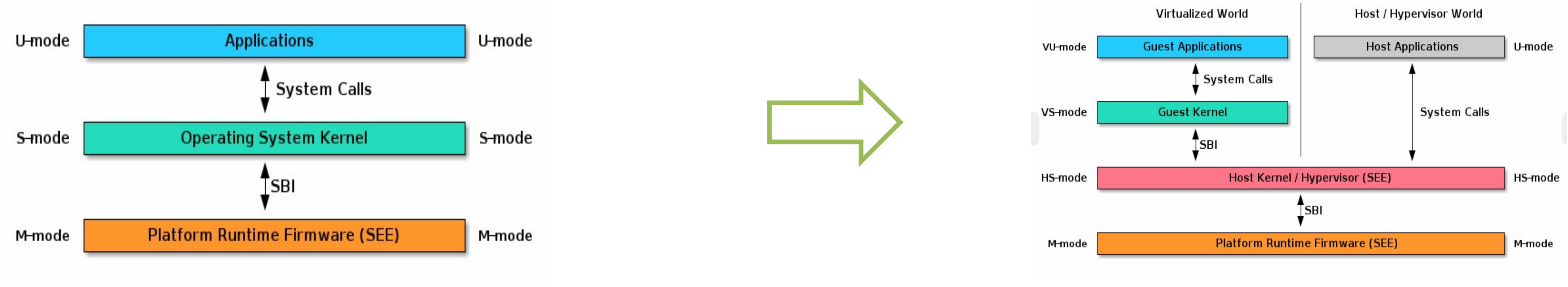
🟧 M-mode Software
🟥 HS-mode Software
🟩 VS-mode Software
🟦 VU-mode Software
⬜ U-mode Software



App1 | App2
ABI | ABI
OS (S-Mode)
SBI
SEE (M-Mode)

□ Suitable for both Type-1 (Baremetal) and Type-2 (Hosted) hypervisors

□ In HS-mode, an OS or hypervisor interacts with the machine through the same SBI as an OS normally does from S-mode. An HS-mode hypervisor is expected to implement the SBI for its VS-mode guest.

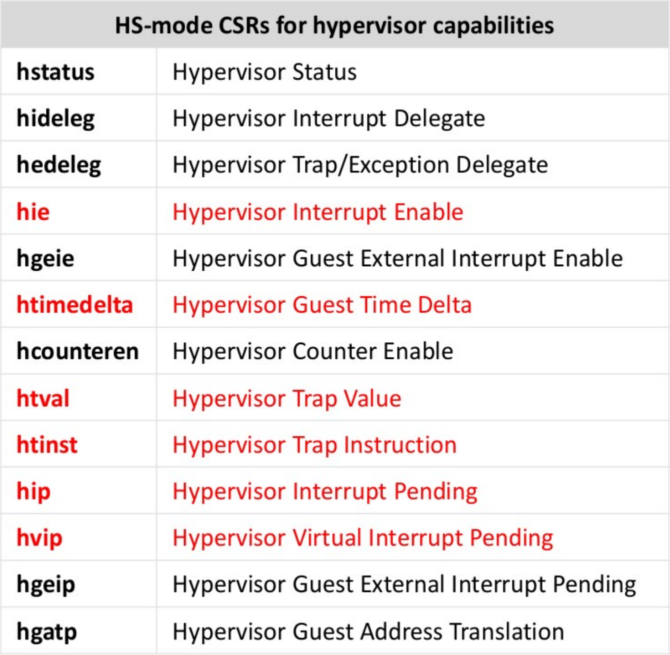


U-mode Applications U-mode
System Calls
S-mode Operating System Kernel S-mode
SBI
M-mode Platform Runtime Firmware (SEE) M-mode

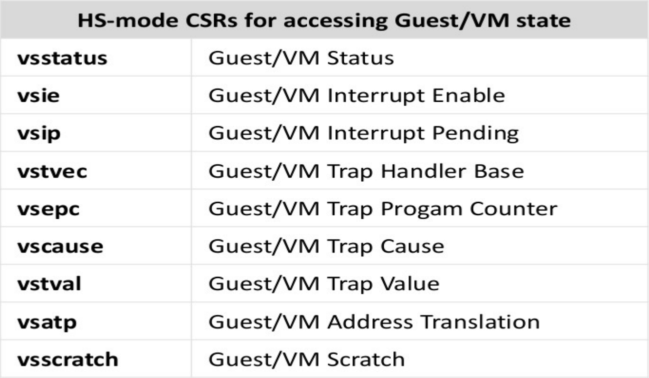
Virtualized World
Host / Hypervisor World
VU-mode Guest Applications
Host Applications U-mode
System Calls
VS-mode Guest Kernel
System Calls
SBI
HS-mode Host Kernel / Hypervisor (SEE) HS-mode
SBI
M-mode Platform Runtime Firmware (SEE) M-mode

RISC-V H-Extension: Hypervisor CSRs

ESWIN



HS-mode CSRs for hypervisor capabilities	
hstatus	Hypervisor Status
hideleg	Hypervisor Interrupt Delegate
hedeleg	Hypervisor Trap/Exception Delegate
hie	Hypervisor Interrupt Enable
hgeie	Hypervisor Guest External Interrupt Enable
htimedelta	Hypervisor Guest Time Delta
hcounteren	Hypervisor Counter Enable
htval	Hypervisor Trap Value
htinst	Hypervisor Trap Instruction
hip	Hypervisor Interrupt Pending
hvip	Hypervisor Virtual Interrupt Pending
hgeip	Hypervisor Guest External Interrupt Pending
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

Modification to machine-level CSR	
misa	Mip/mie
mstatus/mstatush	mtval2
mideleg	mtinst

□ More control registers for virtualising S-mode

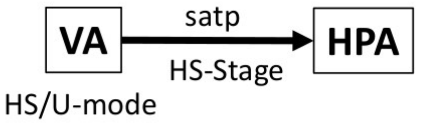
- □ HS-mode (V=0): s<xyz> CSRs point to standard s<xyz> CSRs, h<xyz> CSRs for hypervisor capabilities, vs<xyz> CSRs contains VS-mode state
- □ VS-mode (V=1): s<xyz> CSRs point to virtual vs<xyz> CSRs

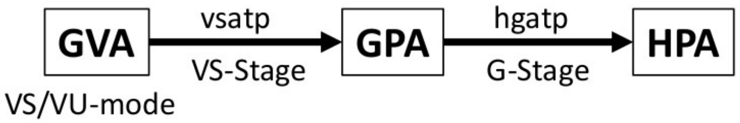
□ HFENCE and HLV/HSV instructions are Hypervisor Instructions

RISC-V H-Extension: MMU

ESWIN

- □ One-Stage MMU for HS/U-mode
  - ◆ HS-mode page table (HS-Stage)
    - ➢Translate hypervisor Virtual Address (VA) to Host Physical Address (HPA)
    - ➢Programmed by Hypervisor using satp CSR
- □ Two-Stage MMU for VS/VU-mode
  - ◆ VS-mode page table (VS-Stage)
    - ➢ Translates Guest Virtual Address (GVA) to Guest Physical Address (GPA)
    - ➢ Programmed by Guest using satp (aka vsatp) CSR
  - ◆ HS-mode guest page table (G-Stage)
    - ➢ Translates Guest Physical Address (GPA) to Host Physical Address (HPA)
    - ➢ Programmed by Hypervisor using hgatp CSR
- □Format of all above page tables is same





Hardware optimised guest memory management

RISC-V H-Extension: MMIO & Interrupts

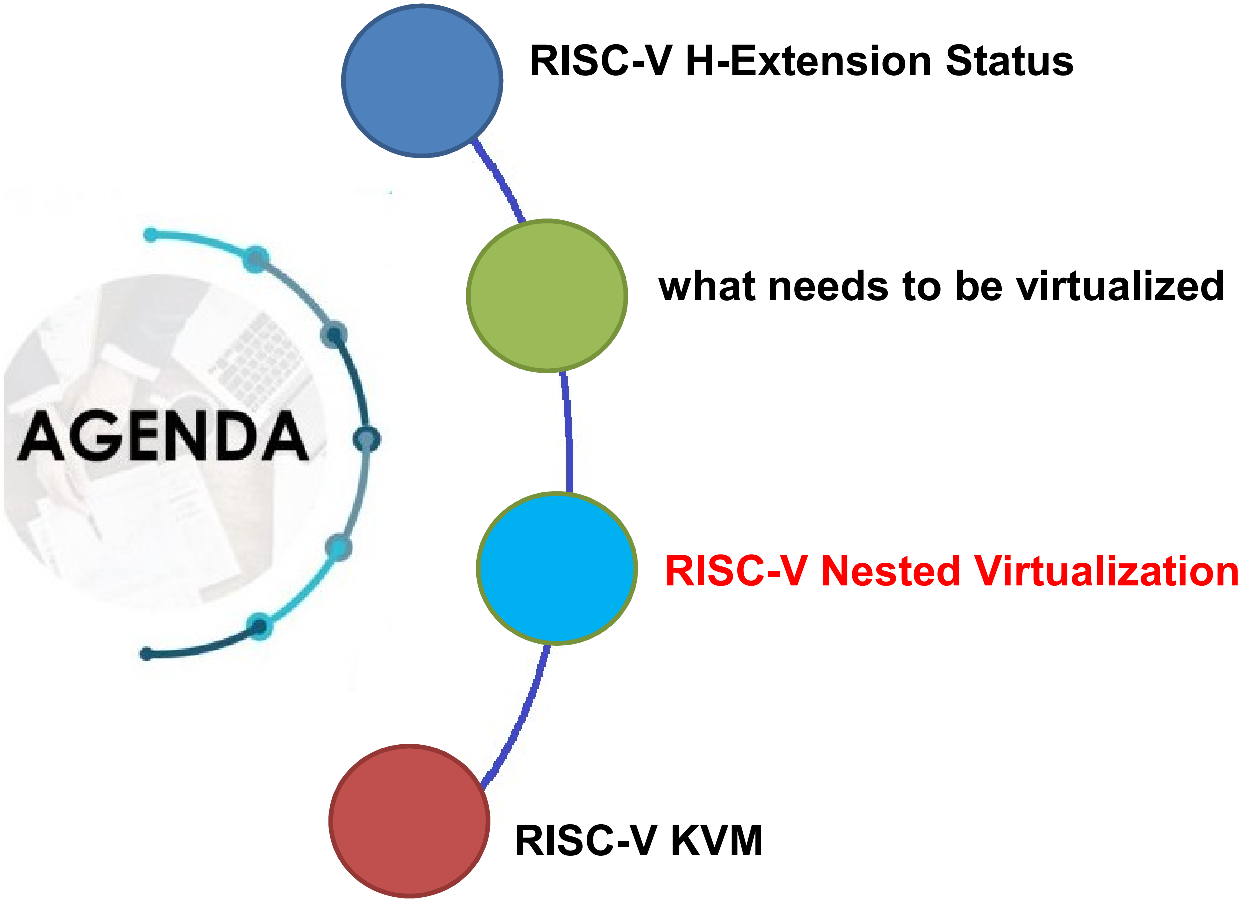
- Guest virtual interrupts are injected by updating hvip CSR from HS-mode
  - hvip.VSEIP bit for Hypervisor injected virtual external interrupt
  - hvip.VSTIP bit for Hypervisor injected virtual timer interrupt
  - hvip.VSSIP bit for Hypervisor injected virtual inter-processor interrupt

- Virtual timer and inter-processor interrupts injected based on SBI calls from Guest

- Hypervisor can trap-n-emulate Guest MMIO using HS-mode guest page table
  - Software emulated PLIC
  - VirtIO devices
  - Other software emulated peripherals

Guest MMIO and Interrupts virtualization

目录



RISC-V H-Extension Status

what needs to be virtualized

RISC-V Nested Virtualization

RISC-V KVM

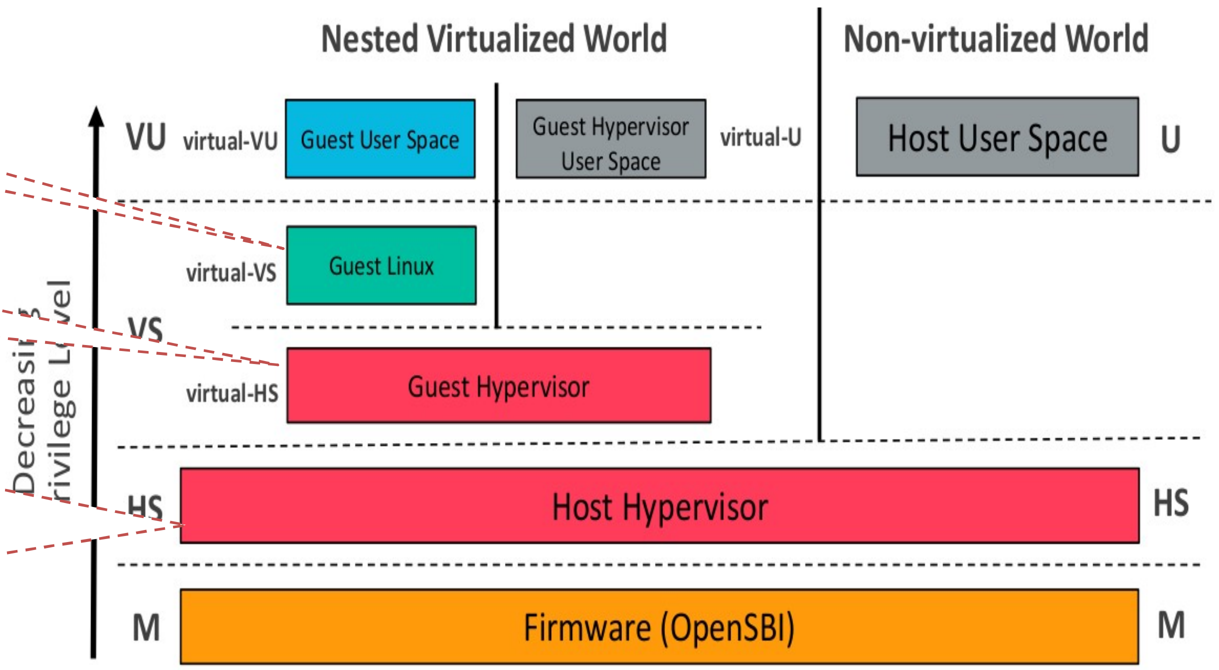
RISC-V Nested Virtualization

ESWIN

Guest OS of guest hypervisor also runs in VS-mode

Guest hypervisor runs in VS-mode, but thinks it's in HS-mode

- ◆ Host hypervisor runs in HS-mode
- ◆ Host hypervisor performs background-foreground CSR swaps on behalf of guest hypervisor
- ◆Host hypervisor maintains shadow page tables



Recursive virtualization supported with additional HS-level software support

目录

ESWIN

RISC-V H-Extension Status

what needs to be virtualized

RISC-V Nested Virtualization

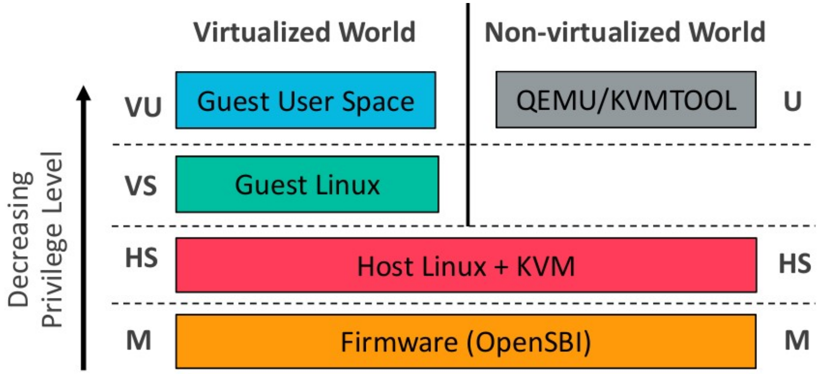
RISC-V KVM

RISC-V KVM Status

ESWIN

The RISC-V port of the KVM hypervisor,Current State:

- Supports H-Extension v0.6.1 draft specification
- No RISC-V specific KVM IOCTL
- Supports both RV32 and RV64 Hosts
- Minimal world-switch and full world-switch via vcpu_load()/vcpu_put()
- Floating point unit lazy save/restore
- KVM ONE_REG interface for user-space
- Timer and IPI emulation in kernel-space
- PLIC emulation is done in user-space
- Hugepage support
- SBI v0.2 interface for Guests
- Unhandled SBI calls forwarded to KVM userspace
- Vhost support using ioeventfd



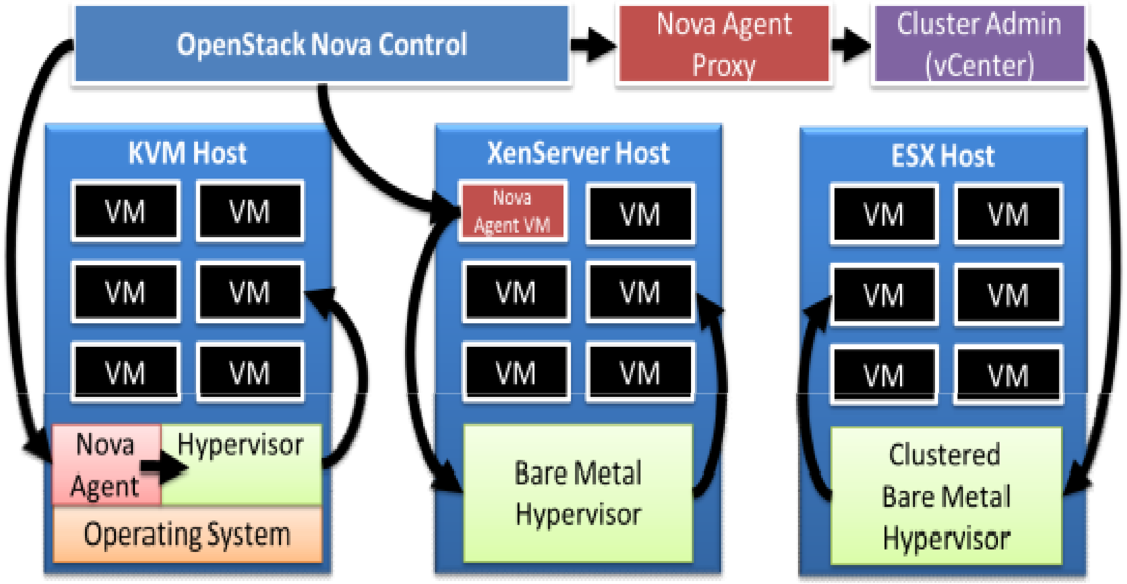
https://github.com/kvm-riscv/linux.git (KVM RISC-V repository. but not Linux native git)

RISC-V KVM : To-do List

□Stage 2 dirty page logging (work already in-progress)
□ Nested virtualization (work already in-progress)
□ Trace points
□ KVM unit test support
□ Virtualize vector extensions
□ Guest/VM migration support
□ Allow 32bit Guests on 64bit Hosts (defined in RISC-V spec)
□ Allow big-endian Guests on little-endian Hosts and vice-versa (defined in RISC-V spec)

Cloud Server Virtualization management - openstack

ESWIN



□ KVM turns the linux kernel into a hypervisor, not user friendly.
□ Openstack is a hypervisor manager, for example, deployment, migration, evacuate instances from the cloud servers.
□ XEN start its porting to RISC-V in 2021