RISC-V Hypervisor Extension

Paolo Bonzini
Red Hat, Inc.

John Hauser
UC Berkeley

Andrew Waterman
SiFive, Inc.

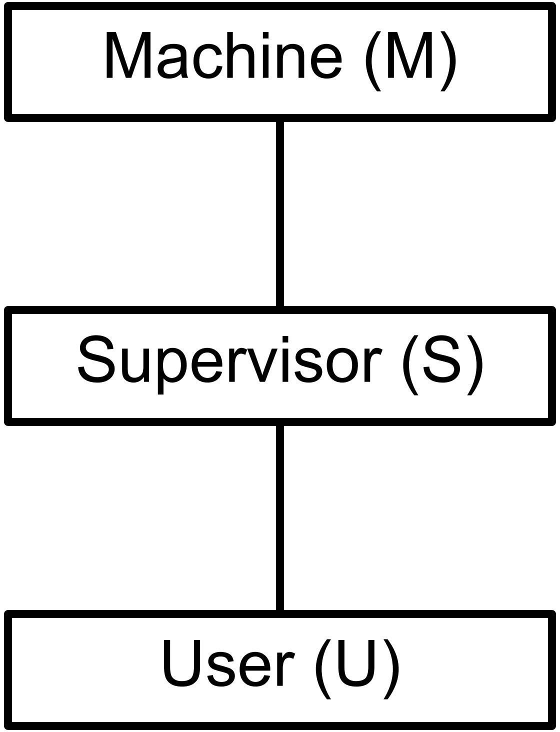
7th RISC-V Workshop
Western Digital, Milpitas, CA
November 28, 2017

Purpose & Goals

- Virtualize S-mode to support running guest OSes under Type-1, Type-2, and hybrid hypervisors
- Support recursive virtualization
- Be performant & parsimonious

RISC-V Privilege Modes

- 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)



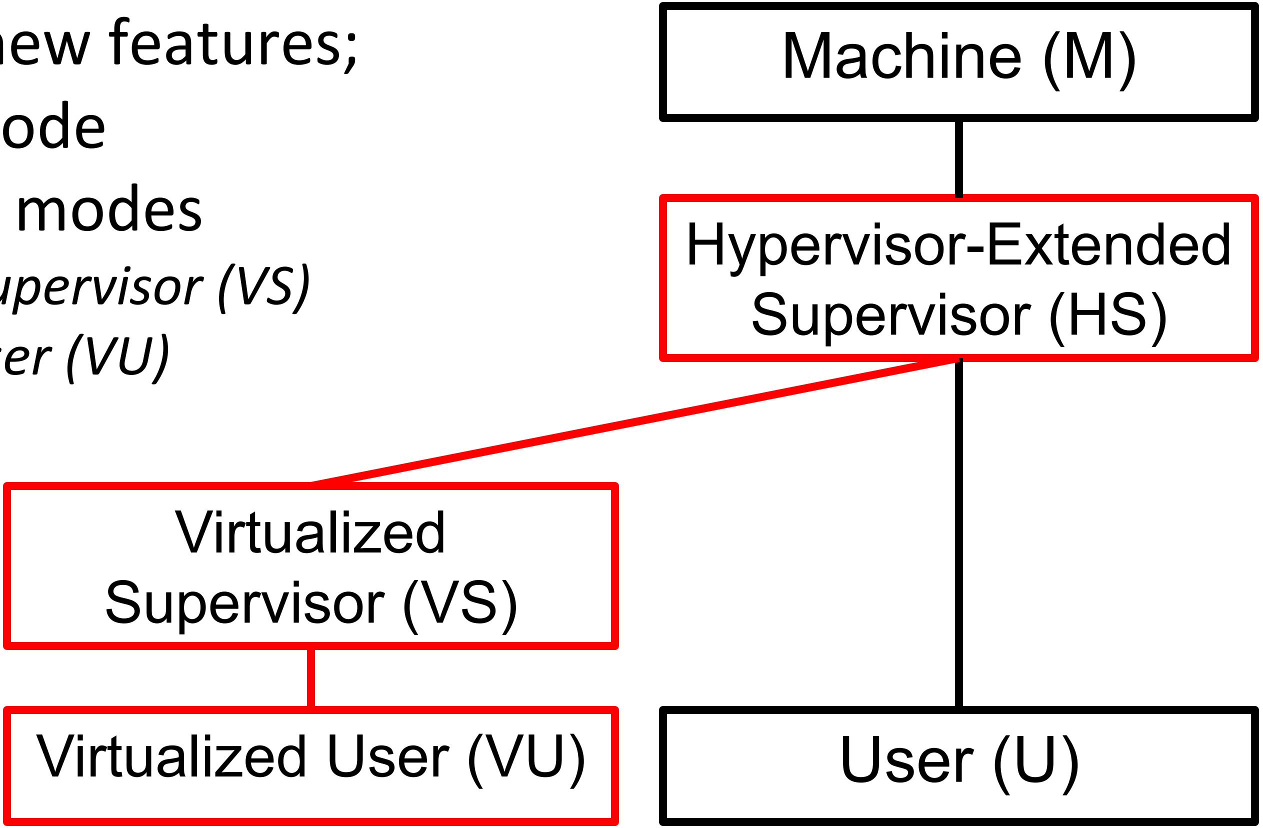
Machine (M)

Supervisor (S)

User (U)

Privilege Modes with Hypervisor Ext.

- S-mode gains new features; becomes HS-mode
- Two additional modes
  - *Virtualized Supervisor (VS)*
  - *Virtualized User (VU)*


Machine (M)
Hypervisor-Extended Supervisor (HS)
Virtualized Supervisor (VS)
Virtualized User (VU)
User (U)

What Needs to be Virtualized?

- Supervisor architectural state (i.e. CSRs)
- Memory
- I/O and Interrupts

Virtualizing Supervisor Arch. State

- Additional copies of most supervisor CSRs (`sscratch`, `sepc`, etc.) provisioned as *background* supervisor CSRs (`bsscratch`, `bsepc`, etc.)
- In HS-mode, foreground CSRs contain HS-mode state; background CSRs contain inactive VS-mode state
- In VS-mode, foreground CSRs contain VS-mode state; background CSRs contain inactive HS-mode state
- Hardware swaps contents of foreground & background CSRs on transition between VU/VS-mode and HS-mode

Virtualizing Memory

- Two-Level Address Translation
  - Original virtual addresses translated to guest physical addresses by VS-level page table
  - Guest physical addresses translated to machine physical addresses by HS-level page table
- Same page-table entry format as S-mode
- Same page-table layouts as S-mode (Sv32, 39, 48, …)

Virtualizing I/O and Interrupts

- Software & Timer interrupts use SBI (=> trivial)
- Use two-level paging scheme to trap MMIO accesses
  - Sufficient to emulate PLIC and other MMIO devices
- Could avoid extra traps into hypervisor with virtualization-aware PLIC
  - Platform issue, outside scope of hypervisor ISA
- Need I/O MMU to initiate DMAs without trap into hypervisor
  - Platform issue, outside scope of hypervisor ISA

Recursive Virtualization

- Recursive virtualization supported with additional HS-level software support
- Host hypervisor runs in HS-mode
- Guest hypervisor runs in VS-mode, but thinks it’s in HS-mode
- Guest OS of guest hypervisor also runs in VS-mode
- Host hypervisor performs background-foreground CSR swaps on behalf of guest hypervisor
- Host hypervisor maintains shadow page tables

Emulating the Hypervisor Extension

- Designed to be efficiently emulatable on M/S/U systems with traps into M-mode
  - SW development can precede hypervisor-capable HW
- Hypervisor runs in S-mode, but thinks it’s in HS-mode
- Guest also runs in S-mode
  - Many fewer emulation traps than classical virtualization
- M-mode TVM feature intercepts page-table operations so M-mode SW can maintain shadow PTs
- M-mode TSR feature intercepts privilege transfers so M-mode SW can swap background CSRs

Implementation Status

- Specification v0.1 available on github
  - https://github.com/riscv/riscv-isa-manual
- Spike implementation Q1 2018, QEMU thereafter
- Need KVM port prior to finalization
  - M-mode emulation layer => lack of silicon not problematic
- Need silicon implementation prior to ratification