bhyve ARM64 PCI Passthrough

15-612 Project Proposal

Joseph Amirtharaj (jamirtha)

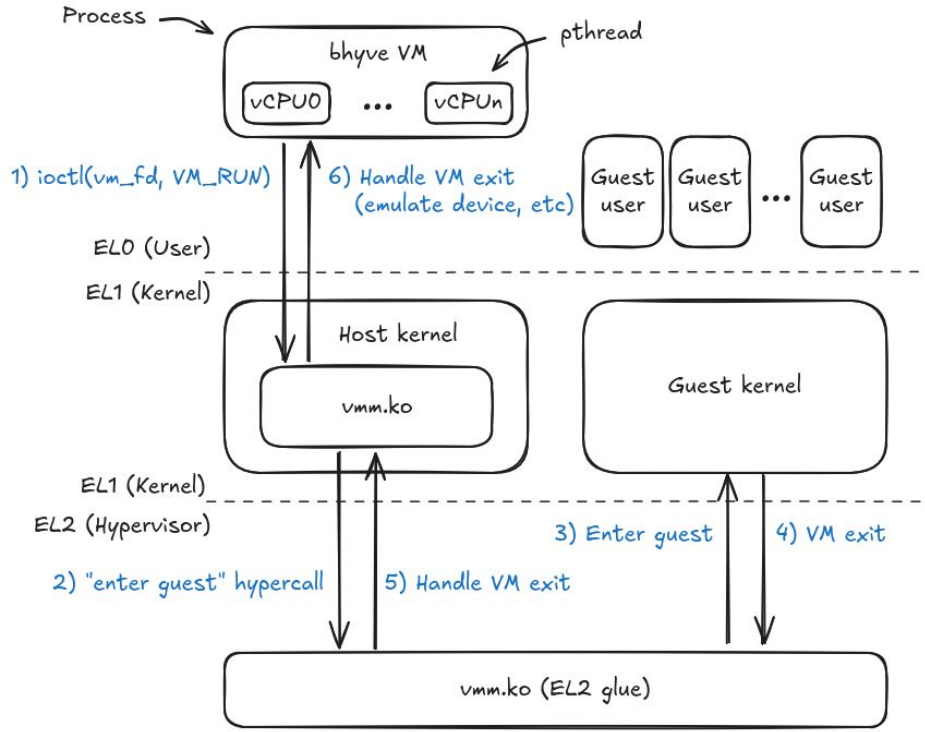
2026-09-11

What is bhyve?

- Short for “BSD Hypervisor”, pronounced “beehive”
- Hypervisor that runs “on top of” FreeBSD
- Runs unmodified guests targeting same architecture as host
- Supports x86-64, ARM64, RISC-V to varying degrees

How does bhyve VM run on ARM64?

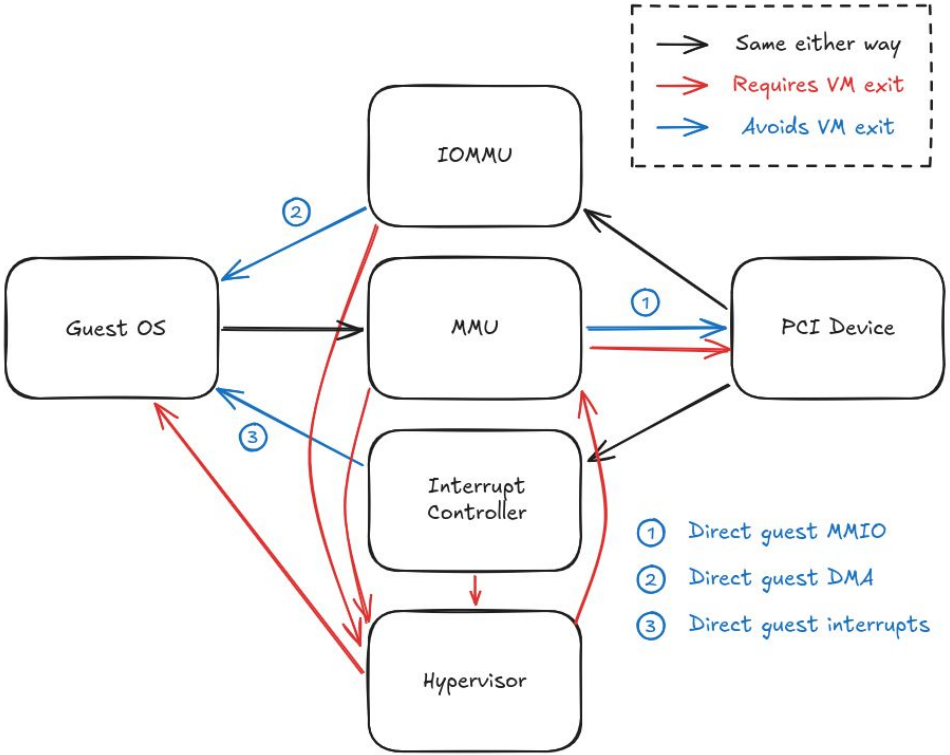
- ARM exception levels:
  - EL0 = User
  - EL1 = Kernel
  - EL2 = Hypervisor
- bhyve VM is EL0 process
- vCPU = thread in bhyve VM
- Each vCPU repeats following:
  - ioctl(vm_fd, VM_RUN)
  - Handle VM exit on return from ioctl
- vmm.ko is bhyve kernel module
  - Contains EL2 glue to mediate between guest and host



Drawn in excalidraw.com

What is PCI passthrough?

- Can emulate PCI device through appropriate VM exits – expensive!
- Pass physical PCI device to VM
  - VM now owns device – even host cannot drive it
- Requirements for PCI passthrough:
  - Enable guest to directly do MMIO through virtual memory
  - Enable DMA to guest's virtual memory using IOMMUs
  - Enable direct virtual interrupt injection



Drawn in excalidraw.com

What work has already been done?

- PCI passthrough supported in bhyve on x86-64 for Intel CPUs with VT-d
- User-space bhyve only emulates accesses to config space + BARs
- PPT driver attaches to device to remove from host IOMMU domain
- Code almost entirely under amd64 paths.
  - sys/amd64/vmm/io/ppt.c: **950 LOC**
  - sys/amd64/vmm/intel/vtd.c: **700 LOC**
  - sys/amd64/vmm/amd/amdvi_hw.c: **1400 LOC**
- Code exists for arm64 SMMU too
  - sys/arm64/iommu/: **5K LOC**

What work needs to be done?

- Find an arm64 system with support for SMMU and GICv4
  - QEMU – ‘virt’ board
  - Used Ampere Altra
- Find simple PCI device to pass through
  - PCI UART
- Write some glue + PPT driver to accomplish PCI passthrough!

Licenses

bhyve: BSD-2-Clause

vmm.ko: BSD-2-Clause

FreeBSD: BSD-2-Clause

Contribution guidelines

CONTRIBUTING.md