redhat.

ARM virt 3.0 and beyond: towards a better scalability

Eric Auger
KVM Forum 2018

Overview

- Mach-virt Introduction
- Scalability Issues
  - VCPU count (qemu 3.0)
  - PCIe bus count (qemu 3.0)
  - RAM size (WIP)
- Conclusion

redhat.

Virt Machine
What it is, what it isn't

- mach-virt does not correspond to real SOC/Board
- Preferred generic machine for running guests compatible with ARMv7/ARMv8 architectures
- Server-class resources:
  - PCIe and virtio-pci ecosystem
  - GICv3, ITS, SMMU
  - Significant amount of CPUs and RAM
- Not fully SBSA compliant (see “SBSA ref QEMU platform” thread)

redhat.

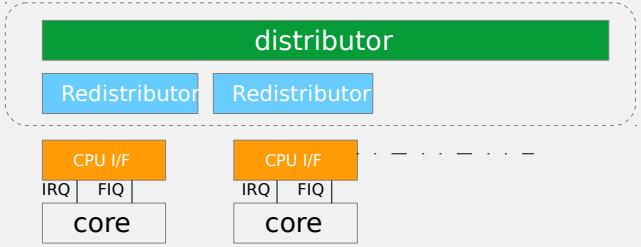
Virt Scalability Issues (< qemu 3.0.0)

- Physical machines
  - hundreds of cores
  - large number of PCIe devices
  - TBs of RAM
- Virt Machine
  - Max 123 vcpus
  - Max 16 Pcie buses
  - Max 255 GB Initial RAM

redhat.

QEMU 3.0: Lift the 123 VCPU Limit

- Limiting single redistributor region
- DT/ACPI allow several discontiguous redistributor regions
- 4.18+ kernel now allows to register multiple VGICv3 redistributor regions
- New 64MB GICv3 redistributor region
  - up to 512 VCPUs

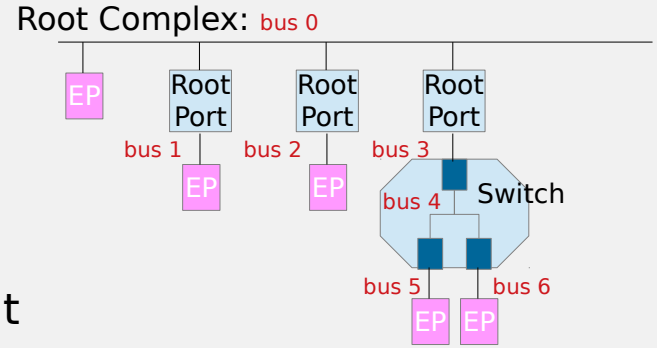


0	128MB	Flash Device	
128MB	256MB	IO Devices Legacy 2*64kB*123 GICV3 rdist region[0]	Original
256MB	1GB	PCIe Host Controller	
1GB	256GB	RAM	
256GB	512GB	64MB rdist region[1]	New
512GB	1TB	High Mem PCIe MMIO Region	

redhat.

QEMU 3.0: Lift the 16 PCIe Bus Limit

- Limiting 16MB ECAM/MMCFG region
- A new 256MB ECAM region (default)
  - not compatible with 32b FW or guest without LPAE (-machine virt,highmem=off)
- More complex topologies and increase in the number of hotpluggable end-points



0	128MB	Flash Device
128MB	256MB	IO Devices
256MB	1GB	PCIe Host Ctrl
		Legacy 16MB ECAM
1GB	256GB	RAM
256GB	512GB	64MB rdist reg[1]
		256MB ECAM region
512GB	1TB	High Mem PCIe MMIO Region

Legacy

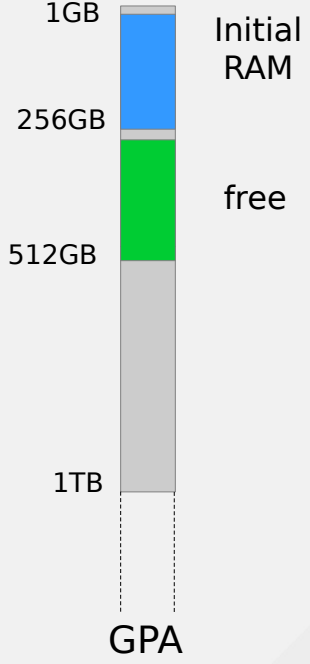
New

redhat.

WIP: Lift the 255GB RAM limit

- Memory optimized VM workloads for large databases/caches, in-memory analytics, ...

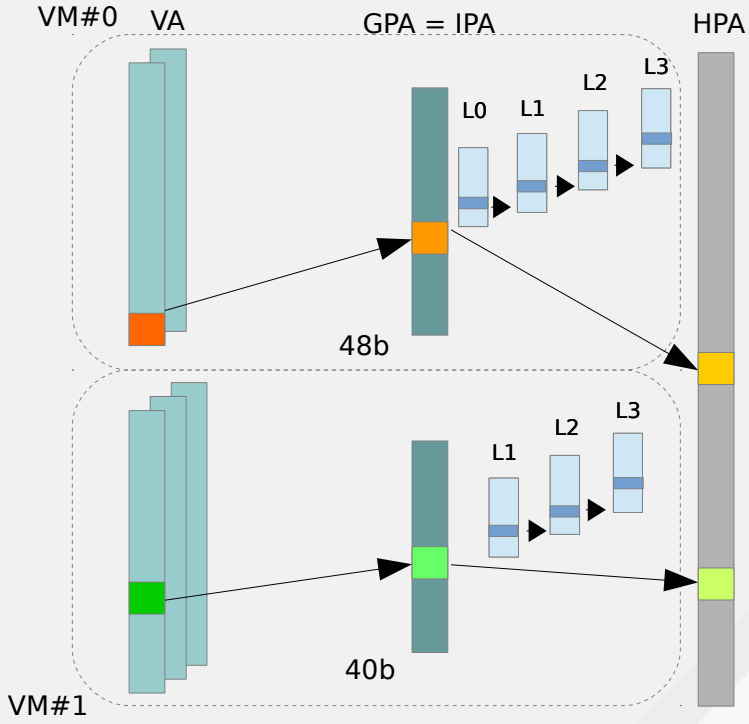
- Lift 40b GPA limit
- Device Memory wanted
- Choose a future-proof layout



redhat.

Lift 40b IPA Limit

- 4.20+: GPA size chosen per VM
  - “kvm: arm64: Dynamic IPA and 52bit IPA”
  - Per VM stage 2 geometry & control register
  - depends on host config and physical CPU (ID_AA64MMFR0_EL1.PARange)
- Still EDK2 ArmVirtQemu also currently limits the PA space to 40 bits by default



redhat.

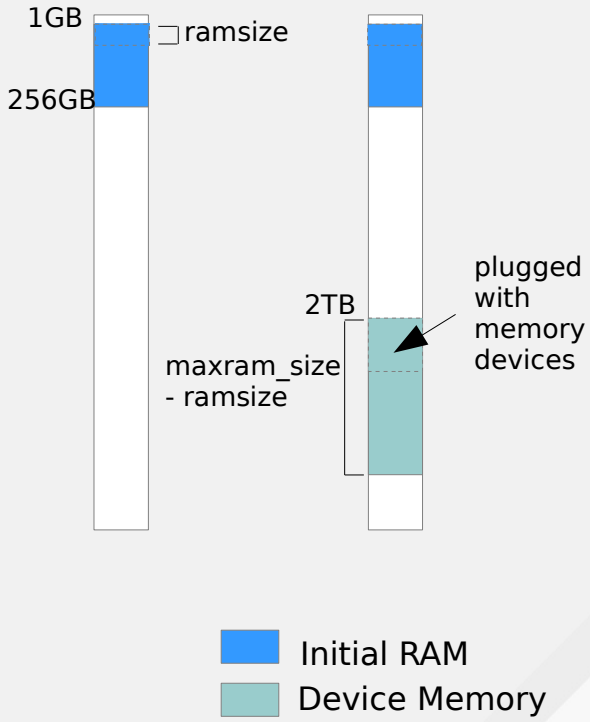
Initial RAM / Device Memory

- Initial Ram (-m)
  - ram_size, allocated at boot time
- Device Memory
  - maxmem=<>[, slots=<>] qemu options
  - maxram_size - ram_size provisioned
  - BE objects/FE memory devices cold-pluggable or hot-pluggable
  - FE devices: pc-dimm, nv-dimm, virtio-mem, virtio-pmem
  - nv-dimm passthrough, DAX on guest, cache page optim, ballooning alternative, ...
- arm64 does not support memory hot-plug/unplug yet
  - [PATCH v2 0/5] Memory hotplug support for arm64 - complete patchset v2 (Nov 17)
  - memory can be cold-plugged though

redhat.

RAM Layout: Split Memory Model

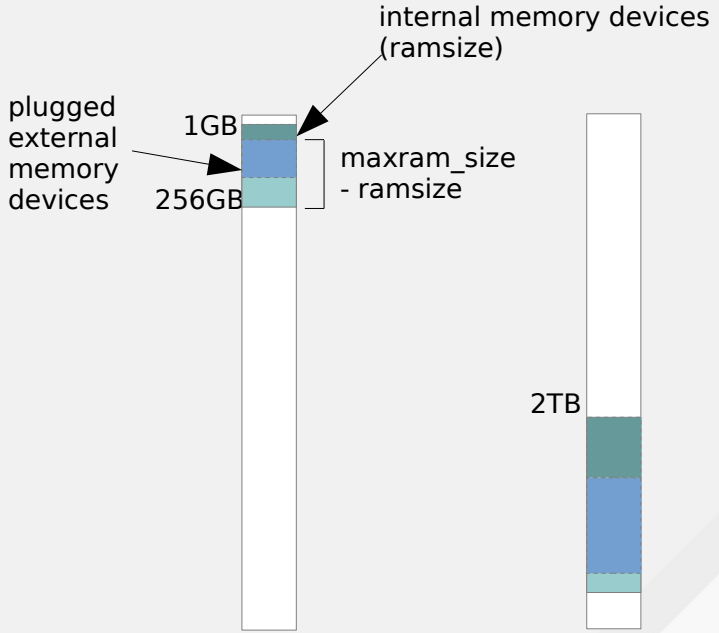
- Legacy initial RAM and new device memory at 2TB
- “ARM virt: PCDIMM/NVDIMM at 2TB” series
- Pros
  - easy QEMU integration
  - no change to the FW
- Cons:
  - initial memory capped at 255GB
  - no device mem below 1TB
  - static memory map (bad experience on PC)



redhat.

RAM Layout: Floating RAM Base

- Single memory region with floating RAM base
- part of a larger refactoring strategy
  - uses only device memory
    - internal memory devices
    - usual external memory devices
- Pros
  - easier to fit/carve out new GPA regions
  - opportunity for NUMA cleanup
- Cons
  - FW impact
  - Larger QEMU impact



redhat.

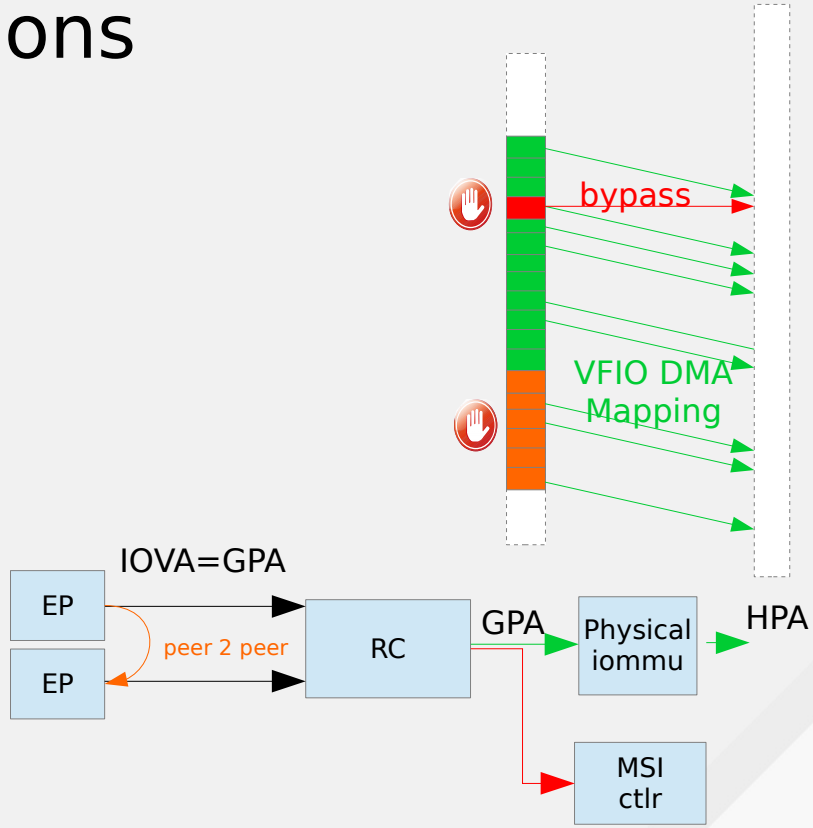
Floating RAM Base Issue

- EDK2 ARMVirtQemu currently assumes
  - DRAM starts at 1GB and starts with DTB
- Floating RAM base → EDK2 rework:
  - Pass info from QEMU to FW
  - Remove hardcoded addresses in FW code
- Decode where the RAM starts:
  - Pass the dtb base address in one reg and parse it in asm as there is no stack (~ 250 LOC, maintenance issue)
  - pass the RAM base address through fw_cfg (~ 50 asm LOC, no stack either). fw_cfg must be at a fixed address though.
  - keep a small 1MB SRAM region used as a temporary RAM for Pre-EFI at 1GB to fit the dtb, used as initial stack. RAM base is parsed from dtb.

redhat.

Host IOVA Reserved Regions

- VFIO maps the whole guest RAM through iommu and guest programs the assigned device for DMA with GPA
- Some GPAs cannot be used:
  - not reaching the IOMMU
  - bypassed by the IOMMU (MSI regions)
  - identify mapping enforced by BIOS (RMRR)
  - reserved for host kernel usage
- QEMU should identify them & carve them out
  - [v6,0/7] vfio/type1: Add support for valid iova list management
  - [RFC v2 0/6] hw/arm: Add support for non-contiguous iova regions
  - GFX/USB RMRR integration to be reworked



redhat.

Conclusion

- VCPU count and PCIe topology Improvements
- WIP related to RAM expansion
  - Rework of the memory layout
  - Floating RAM base experiment
  - Single device memory region experiment
  - New memory device use cases for ARM
- ARM still does not offer
  - VCPU hotplug
  - Memory hotplug
- Host IOVA reserved regions not properly handled

redhat.

References / Credits

- [PATCH v6 00/18] kvm: arm64: Dynamic IPA and 52bit IPA
  - Suzuki K Poulose, ARM, Sept 2018
- [v6,0/7] vfio/type1: Add support for valid iova list management
  - Shameerali Kolothum Thodi, Huawei, April 2018
- [RFC v2 0/6] hw/arm: Add support for non-contiguous iova regions
  - Shameerali Kolothum Thodi, Huawei, May 2018
- [RFC PATCH 0/3] add nvdimm support on AArch64 virt platform
  - Kwangwoo Lee, SK group, July 2016
- [PATCH v2 0/5] Memory hotplug support for arm64 - complete patchset v2
  - Andrea Reale, Maciej Bielski, Scott Branden, Nov 2017
- [RFC v4 00/16] ARM virt: PCDIMM/NVDIMM at 2TB
  - Eric Auger, Red Hat, Oct 2018

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHat

youtube.com/user/RedHatVideos