Agenda

SBSA Reference QEMU Platform
- Purpose
- Features
- Firmware Porting
- Current Status
Agenda

SBSA Reference QEMU Platform
- Purpose
- Features
- Firmware Porting
- Current Status
A platform needed

- Like real arm64 server hardware, as faithful as possible
- Test bed for firmware prototype development
- To install and run an OS release easily
- Simple command to launch is also a hope
- Not to fully implement the SBSA specification
Why not ‘virt’ machine

- It comes with para-virtualized virtio devices
- Not like real arm64 hardware, e.g. AHCI/EHCI devices should be system memory mapped
- Typically suitable for running workloads, not firmware development
- Command parameters and readconfig not suitable for adapting ‘virt’ to want is needed
Agenda

SBSCA Reference QEMU Platform
- Purpose
- Features
- Firmware Porting
- Current Status
SBSA-ref QEMU Features

- Based on the ‘virt’ machine
- EL2 and EL3 enabled by default
- A new memory map, main memory beyond 4G
- Memory mapped AHCI (a.k.a. SATA) controller
- Memory mapped EHCI (a.k.a. USB) controller
- USB mouse and keyboard
- E1000E NIC attached to PCIE bus
- A VGA display on PCIE bus
- GIC version 3
SBSA-ref QEMU Features

- No virtio devices
- No ‘platformbus’ device
- No ‘fw_cfg” device
- QEMU does not supply ACPI table
- QEMU does not offer SMBOIS
- QEMU should supply some DT nodes
Agenda

SBSA Reference QEMU Platform
- Purpose
- Features
- Firmware Porting
- Current Status
ARMTF Porting

- New sbsa platform files created
- GICv3 support added
- BL1 and FIP can be placed in secure Flash0 passed to QEMU
  - This is done at UEFI compiling stage
  - Command option -semihosting not needed any longer
  - UEFI is put into Flash1 accordingly
UEFI Porting

- New sbsa platform files created
- Add support for newly introduced devices
- Adapt to the new memory layout
- Compose EFI flash images with ARMTF images
  - Flash0: ARMTF BL1 and FIP (BL2 + BL31)
  - Flash1: EFI code and EFI variables
- Add SMBIO and ACPI tables (to be done)
Install and run OS

● To run sbsa-ref machine
  ○ qemu-system-aarch64 -M sbsa-ref
     -pflash SBSA_FLASH0.fd -pflash SBSA_FLASH1.fd
     -hda disk-image.qcow [-cdrom os-release.iso]

● Vs to run ‘virt’ machine
  ○ qemu-system-aarch64 -machine virt,secure=on,virtualization=on
     -cpu cortex-a57 -smp 2 -m 1024
     -bios bl1.bin -semihosting-config target=native
     -device virtio-scsi-device,id=scsi
     -drive file=disk-image.qcow,id=rootimg,if=none -device scsi-hd,drive=rootimg
     -netdev user,id=unet -device virtio-net-device,netdev=unet -net user
        [-drive file=os-release.iso,id=cddrive,if=none -device scsi-cd,drive=cddrive]
Agenda

SBSA Reference QEMU Platform
- Purpose
- Features
- Firmware Porting
- Current Status
Status Today

- **Patch v2 upstream discussed**
  - Design purpose and intended usage
  - Probably name it ‘sbsa-ref’
  - Should be in a separated C file
  - Not supply ACPI, device tree nodes to be confirmed

- **Patch v3 upstream in progress**

- **Codes also can be found**
  - [http://git.linaro.org/people/hongbo.zhang/qemu-enterprise.git/log/?h=sbsa-upstream-v3](http://git.linaro.org/people/hongbo.zhang/qemu-enterprise.git/log/?h=sbsa-upstream-v3)
Other Issues

- PSCI conduit passing in device tree
  - This needs to be generated from ARMTF, and passed to UEFI via device tree
- MPIDR isn’t emulated well
  - PSCI needs to check it while booting up secondary cores
- Have to temporarily revert some commits to boot
  - Revert "device_tree: Increase FDT_MAX_SIZE to 1 MiB"
  - Revert "target/arm: Implement new do_transaction_failed hook"
Thank You