Linaro Connect
MADRID 2024 | MAY 14-17 2024

Gunyah Accelerator for Qemu

Srivatsa Vaddagiri
Principal Engineer, Qualcomm India Private Limited

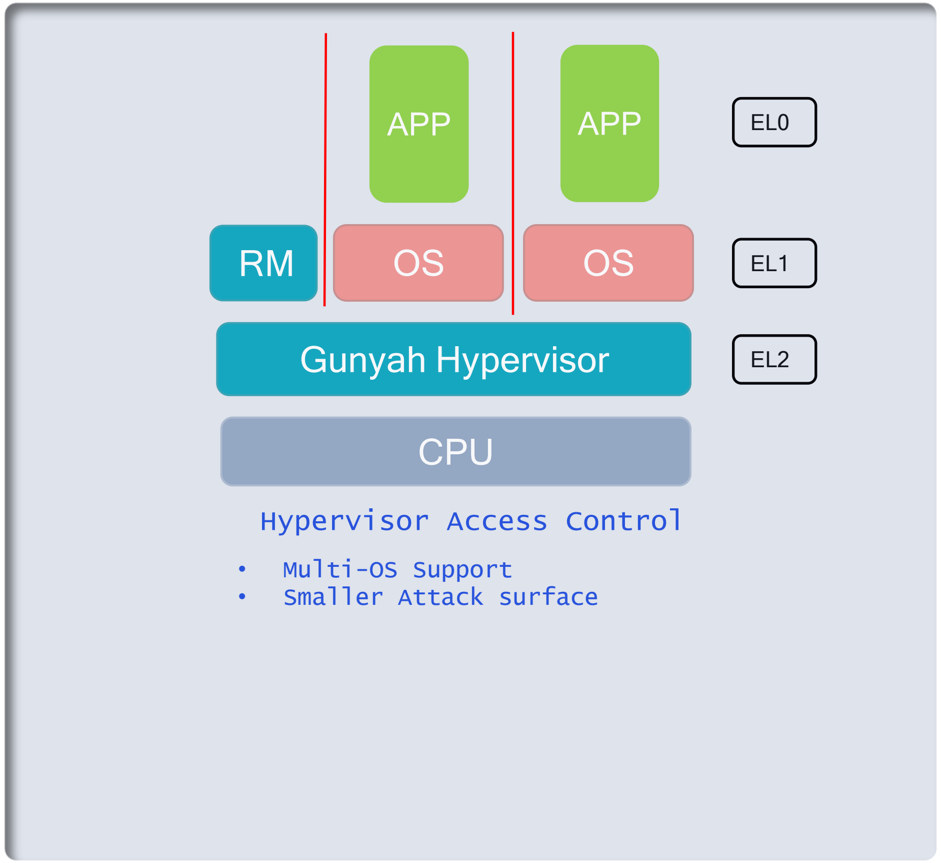
Agenda

- Gunyah Hypervisor
- Gunyah Linux kernel driver
- Qemu support

What is Gunyah™ Hypervisor Software ?

- Hypervisor solution implemented by Qualcomm Technologies, Inc.
- EL2 Hypervisor is small microkernel
- “Resource Manager” VM implements policy for EL2 & runs isolated from other VMs

Getting the source:
github.com/quic/gunyah-hypervisor



Gunyah Key Features

- VM types supported
  - Confidential VMs – Guest memory is protected from host
  - Untrusted VMs – Guest memory can be accessed by host
- Confidential VM types:
  - Trusted VM
    - Hypervisor ***enforces*** VM image authentication by Qualcomm Trustzone ***before*** letting VM start
    - VM image include device-tree
  - Google VM
    - VM image authentication outside scope of hypervisor
    - Typically authenticated by PVM firmware – a software blob that runs first as part of VM (before main image)

Gunyah Key Features

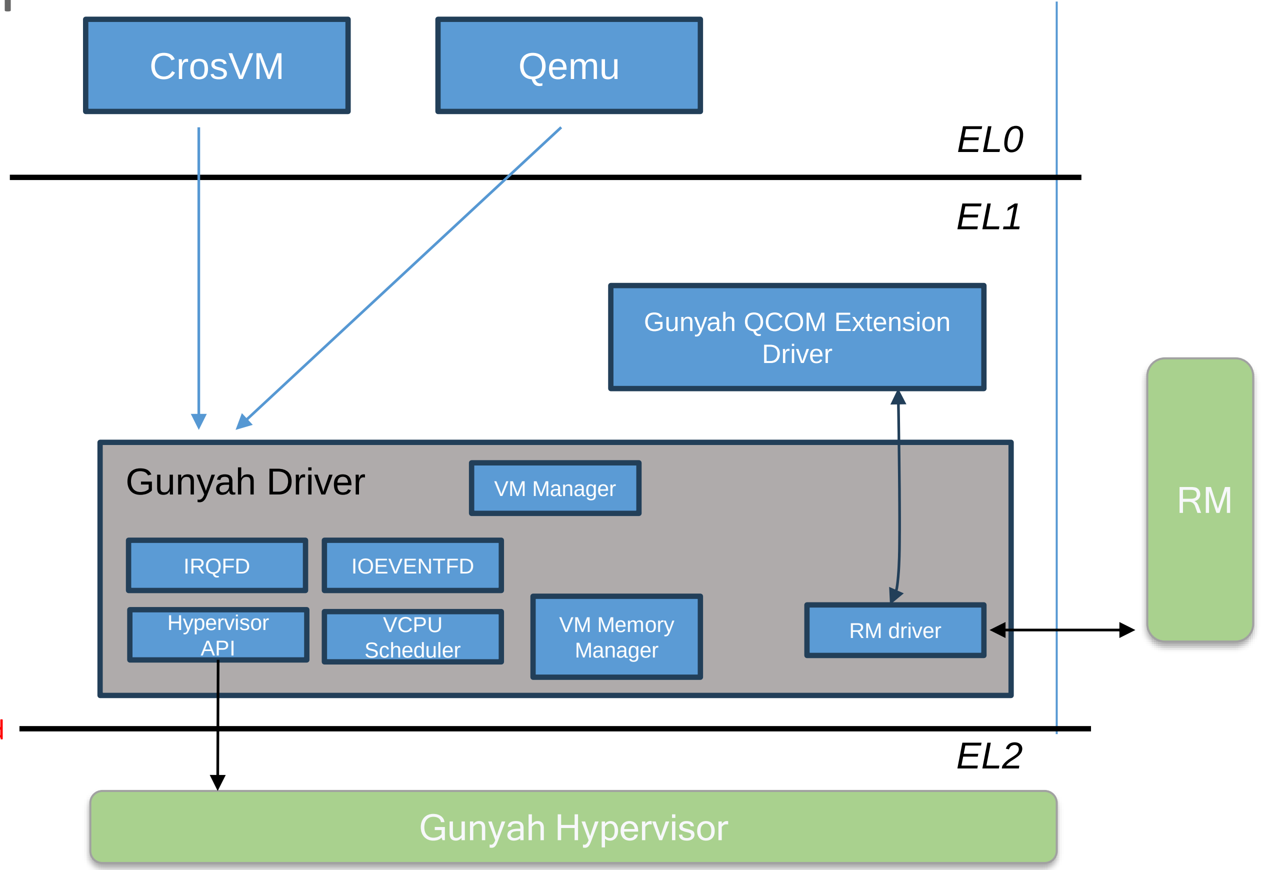
- VM types supported
  - Confidential VMs – Guest memory is protected from host
  - Untrusted VMs – Guest memory can be accessed by host
- Confidential VM types:
  - Trusted VM
    - Hypervisor ***enforces*** VM image authentication by Qualcomm Trustzone ***before*** letting VM start
    - VM image include device-tree
  - Google VM
    - VM image authentication outside scope of hypervisor
    - Typically authenticated by PVM firmware – a software blob that runs first as part of VM (before main image)
- Confidential VMs
  - VM memory is private (isolated from an untrusted host via ***page-table based protection***)
  - Additional shared memory possible
    - Memory shared between host and guest
    - Shared memory to be assigned by host ***before*** guest starts
    - No runtime API for guest to share its private memory
  - Interrupt virtualization at EL2
  - SMMU based protection from malicious devices
  - VCPU scheduler – both proxy and hypervisor native mechanisms supported
  - MMIO windows (0 - 1GB is allowed for emulation)
  - Wipe memory at warm reset

Gunyah – Key features (contd ..)

- Pre-host VMs
- Inter VM communication
  - Shared Memory
  - Doorbell
  - Message Queue
- Demand Paging of VM memory
- Device Passthrough
- Meet automotive requirements
- Performance optimized for mobile/auto/IoT use cases

Linux Gunyah Driver

- Assists a VMM in VM management functions
- V17 posted upstream by Elliot Berman
- UAPI:
    - Create VM, VCPU
    - RUN VCPU
    - Register eventfd for IRQ injection or notification of IO access by VM
    - Start VM
    - Share or Lend memory to VM. Lending supported by driver variant in ACK
    - Specify Device Tree location
    - Set boot VCPU's initial register context
    - Set Firmware Configuration (Android specific)
- Future changes?
    - Support for additional VM types (Trusted VMs and untrusted VM)
    - Device Assignment



Gunyah Accelerator for Qemu

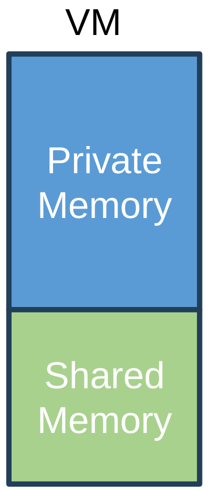
- Work in progress (not merged yet). V2 RFC patches posted
- Supports bring up VM (both confidential and unprotected types). Virtio-PCI devices have been tested.
  - Tested on both Qualcomm SoC and Qemu virtual platform (running open-source Gunyah)
- Supported only for AARCH64 target

```
./qemu-system-aarch64 -machine virt  --accel gunyah …
```

- VM Creation = GH_CREATE_VM
- VCPU creation = GH_VM_ADD_FUNCTION(GH_FN_VCPU, …id=vcpu_id)
- arm virt machine Changes
  - Confidential Guest support
  - Device Tree Customization

Confidential Guests

- Hypervisor-assisted confidential guests
- Guest memory – split into private portion and optionally a shared portion
    - Private portion – memory not accessible by host
        - Used for Guest kernel and application text/data
    - Shared portion – memory shared with host
        - Data that needs to be shared with host (ex: virtio)
- Hypervisor guarantees that the private portion is not visible to host (page-table based protection)
- Optional parameter, ***swiotlb-size***, specifies the shared portion size
- Device Tree changes:
    - Add "/reserved-memory/restricted_dma_reserved" node whose size/reg property indicates *swiotlb-size*
    - Compatible = restricted-dma-pool



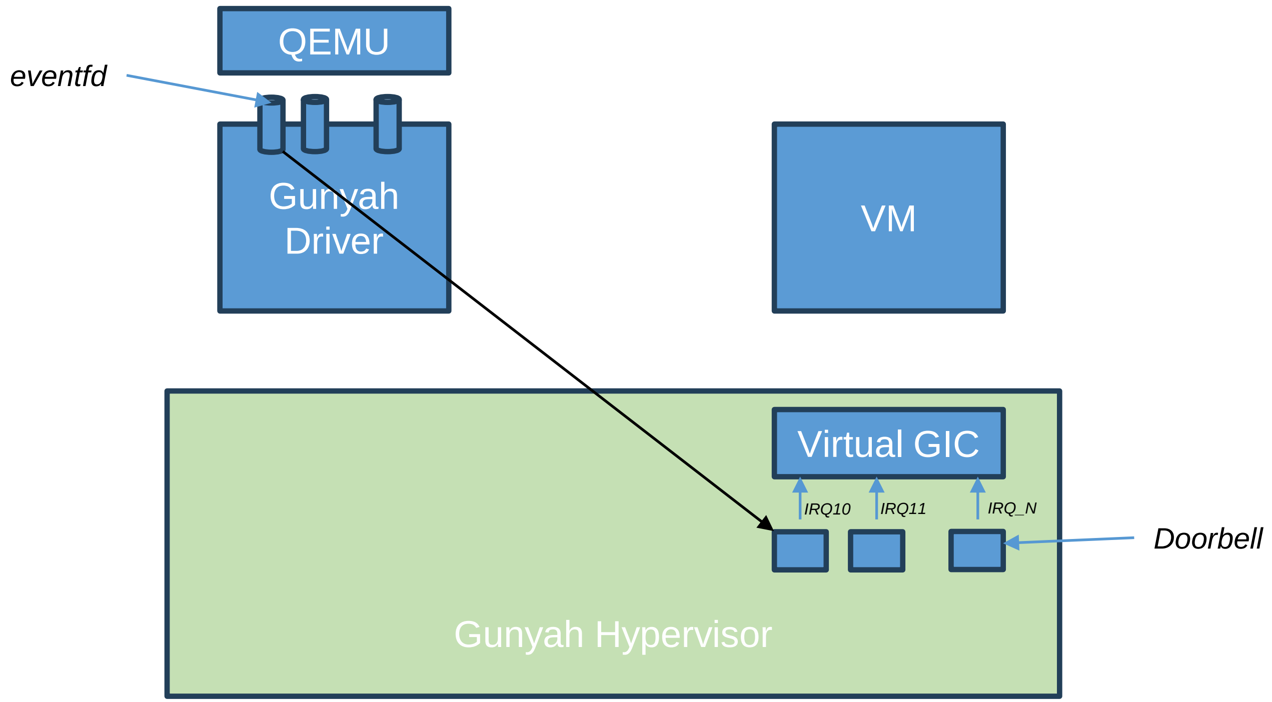
Memory Assignment

- All of VM's memory need to be assigned ***before*** it begins execution
- Memory can be LENT or SHARED
    - LENT memory is made private to guest
    - SHARED memory is made shared between guest and host
- No API (at this time) for guest to share part of its private memory with host
    - Any shared memory required needs to be assigned to guest before it starts
    - Guest needs to be told where in its address space shared memory can be found
- Non-confidential guests : All memory is SHARED
- Confidential guests :
    - (*ram_size – swiotlb_size*) is LENT
    - *swiotlb_size* is SHARED

Scheduling VCPUs

- Gunyah hypervisor supports both proxy and native (vcpu) scheduler
- Gunyah accelerator of Qemu currently supports only proxy scheduled VMs
  - Supporting hypervisor-scheduled VMs is a matter of adding additional DT nodes and VCPU related ioctls.
  - May be supported in future
- Proxy scheduling = Donate thread's time to a VCPU of VM
  - VCPU_RUN ioctl -> *GH_HYPERCALL_VCPU_RUN* hypercall
  - Return value could indicate:
    - MMIO access (device emulation in Qemu)
    - VM exit

Interrupt Controller



Doorbell

- Virtual devices that can be associated with an interrupt
- *doorbell_send* hypercall API can be used by host VM to request injection of associated interrupt
- Doorbell created and bound to an interrupt via DT:
- An eventfd can be bound to a doorbell.

```
gunyah-vm-config {

    vdevices {

        dbl-1 {
            vdevice-type = "doorbell";
            generate = "/hypervisor/dbl-1";
            qcom,label = <0x01>;
            peer-default;
            source-can-clear;
            interrupts = <0x00 0x01 0x04>;
        }
```

```
struct gh_fn_irqfd_arg ghirqfd;

fdesc.type = GH_FN_IRQFD;
fdesc.arg_size = sizeof(struct gh_fn_irqfd_arg);
fdesc.arg = (__u64)(&ghirqfd);

ghirqfd.label = X;  // label (X) represents interrupt number
ghirqfd.fd = irqfd;    // @irqfd eventfd is bound to interrupt X
ghirqfd.flags = GH_IRQFD_FLAGS_LEVEL;

ret = gunyah_vm_ioctl(GH_VM_ADD_FUNCTION, &fdesc);
```

- Injecting an interrupt is a matter of writing to associated eventfd

Interrupt Controller

- Gunyah hypervisor emulates GICv3 for VMs
- ITS not supported
- Key attributes of GICv3 (like the address for redistributor/distributor registers) conveyed via DT, which is interpreted by RM before VM starts
- Each SPI is associated with a doorbell and eventfd.
- Each *eventfd* registered with Linux driver for a specific doorbell
- Qemu can inject a specific interrupt by writing to the associated eventfd

Future Work

- Consolidate confidential VM changes with KVM
- Tracing
- Updates based on kernel UAPI changes before seeking merge
- Device Assignment
- Continuous Integration Tests enabled for Gunyah

Questions ? quic_svaddagi@quicinc.com

Linaro Connect
MADRID 2024 | MAY 12-17 2024

Thank you

Device Assignment

- Exploring VFIO framework
- Some challenges for secure device assignment:
  - Device Attestation
  - Device and related resources are all assigned as unit
  - Handling IOMMU topology changes at runtime
  - Device sanitization after VM crash
  - Multi-VM assignment

Gunyah Support in Qemu

- CMDLine changes
- Scheduler
- Memory Management
- Interrupt Controller
- Device Tree related
- Starting VM (Boot CPU Registers)
- Run Loop
- Virtio Devices
- Generic changes introduced in arm machine
    - Confidential guest support
        - Swiotlb
        - Memory reservation (dma pool)
        - MMIO windows
    - DTB modify