POWER9 eXternal Interrupt Virtualization Engine (XIVE Gen1)

Device types supported:

- KVM_DEV_TYPE_XIVE POWER9 XIVE Interrupt Controller generation 1

This device acts as a VM interrupt controller. It provides the KVM interface to configure the interrupt sources of a VM in the underlying POWER9 XIVE interrupt controller.

Only one XIVE instance may be instantiated. A guest XIVE device requires a POWER9 host and the guest OS should have support for the XIVE native exploitation interrupt mode. If not, it should run using the legacy interrupt mode, referred as XICS (POWER7/8).

- Device Mappings

  The KVM device exposes different MMIO ranges of the XIVE HW which are required for interrupt management. These are exposed to the guest in VMAs populated with a custom VM fault handler.

  1. Thread Interrupt Management Area (TIMA)

  Each thread has an associated Thread Interrupt Management context composed of a set of registers. These registers let the thread handle priority management and interrupt acknowledgment. The most important are :

    - Interrupt Pending Buffer (IPB)
    - Current Processor Priority (CPPR)
    - Notification Source Register (NSR)

  They are exposed to software in four different pages each proposing a view with a different privilege. The first page is for the physical thread context and the second for the hypervisor. Only the third (operating system) and the fourth (user level) are exposed the guest.

  2. Event State Buffer (ESB)

  Each source is associated with an Event State Buffer (ESB) with either a pair of even/odd pair of pages which provides commands to manage the source: to trigger, to EOI, to turn off the source for instance.

  3. Device pass-through

  When a device is passed-through into the guest, the source interrupts are from a different HW controller (PHB4) and the ESB pages exposed to the guest should accommadate this change.

  The passthru_irq helpers, kvmppc_xive_set_mapped() and kvmppc_xive_clr_mapped() are called when the device HW irqs are mapped into or unmapped from the guest IRQ number space. The KVM device extends these helpers to clear the ESB pages of the guest IRQ number being mapped and then lets the VM fault handler repopulate. The handler will insert the ESB page corresponding to the HW interrupt of the device being passed-through or the initial IPI ESB page if the device has being removed.

  The ESB remapping is fully transparent to the guest and the OS device driver. All handling is done within VFIO and the above helpers in KVM-PPC.

- Groups:

1. KVM_DEV_XIVE_GRP_CTRL
   Provides global controls on the device

Attributes:

1.1 KVM_DEV_XIVE_RESET (write only) Resets the interrupt controller configuration for sources and event queues. To be used by kexec and kdump.

Errors: none

1.2 KVM_DEV_XIVE_EQ_SYNC (write only) Sync all the sources and queues and mark the EQ pages dirty. This to make sure that a consistent memory state is captured when migrating the VM.

Errors: none

1.3 KVM_DEV_XIVE_NR_SERVERS (write only) The kvm_device_attr.addr points to a __u32 value which is the number of interrupt server numbers (ie, highest possible vcpu id plus one).

Errors:

-EINVAL	Value greater than KVM_MAX_VCPU_IDS.
-EFAULT	Invalid user pointer for attr->addr.
-EBUSY	A vCPU is already connected to the device.

2. KVM_DEV_XIVE_GRP_SOURCE (write only)
   Initializes a new source in the XIVE device and mask it.

Attributes:
   Interrupt source number (64-bit)

The kvm_device_attr.addr points to a __u64 value:

```
bits:     | 63   ....  2 |   1   |   0
values:   |    unused    | level | type
```

- type: 0:MSI 1:LSI
- level: assertion level in case of an LSI.

Errors:

-E2BIG	Interrupt source number is out of range
-ENOMEM	Could not create a new source block
-EFAULT	Invalid user pointer for attr->addr.
-ENXIO	Could not allocate underlying HW interrupt

3. KVM_DEV_XIVE_GRP_SOURCE_CONFIG (write only)
   Configures source targeting

Attributes:
   Interrupt source number (64-bit)

The kvm_device_attr.addr points to a __u64 value:

```
bits:     | 63   ....  33 |  32  | 31 .. 3 |  2 .. 0
values:   |    eisn       | mask |  server | priority
```

- priority: 0-7 interrupt priority level
- server: CPU number chosen to handle the interrupt
- mask: mask flag (unused)
- eisn: Effective Interrupt Source Number

Errors:

-ENOENT	Unknown source number
-EINVAL	Not initialized source number
-EINVAL	Invalid priority
-EINVAL	Invalid CPU number.
-EFAULT	Invalid user pointer for attr->addr.
-ENXIO	CPU event queues not configured or configuration of the underlying HW interrupt failed
-EBUSY	No CPU available to serve interrupt

4. KVM_DEV_XIVE_GRP_EQ_CONFIG (read-write)
   Configures an event queue of a CPU

Attributes:
   EQ descriptor identifier (64-bit)

The EQ descriptor identifier is a tuple (server, priority):

```
bits:     | 63   ....  32 | 31 .. 3 |  2 .. 0
values:   |    unused     |  server | priority
```

The kvm_device_attr.addr points to:

```
struct kvm_ppc_xive_eq {
    __u32 flags;
    __u32 qshift;
    __u64 qaddr;
    __u32 qtoggle;
    __u32 qindex;
    __u8  pad[40];
};
```

- flags: queue flags

  KVM_XIVE_EQ_ALWAYS_NOTIFY (required)
     forces notification without using the coalescing mechanism provided by the XIVE END ESBs.

- qshift: queue size (power of 2)

- qaddr: real address of queue
- qtoggle: current queue toggle bit
- qindex: current queue index
- pad: reserved for future use

Errors:

-ENOENT	Invalid CPU number
-EINVAL	Invalid priority
-EINVAL	Invalid flags
-EINVAL	Invalid queue size
-EINVAL	Invalid queue address
-EFAULT	Invalid user pointer for attr->addr.
-EIO	Configuration of the underlying HW failed

5. KVM_DEV_XIVE_GRP_SOURCE_SYNC (write only)
   Synchronize the source to flush event notifications

Attributes:
  Interrupt source number (64-bit)

Errors:

-ENOENT	Unknown source number
-EINVAL	Not initialized source number

- VCPU state

  The XIVE IC maintains VP interrupt state in an internal structure called the NVT. When a VP is not dispatched on a HW processor thread, this structure can be updated by HW if the VP is the target of an event notification.

  It is important for migration to capture the cached IPB from the NVT as it synthesizes the priorities of the pending interrupts. We capture a bit more to report debug information.

  KVM_REG_PPC_VP_STATE (2 * 64bits):

```
bits:     |  63  ....  32  |  31  ....  0  |
values:   |   TIMA word0   |   TIMA word1  |
bits:     | 127       ..........       64  |
values:   |            unused              |
```

- Migration:

  Saving the state of a VM using the XIVE native exploitation mode should follow a specific sequence. When the VM is stopped :

  1. Mask all sources (PQ=01) to stop the flow of events.

  2. Sync the XIVE device with the KVM control KVM_DEV_XIVE_EQ_SYNC to flush any in-flight event notification and to stabilize the EQs. At this stage, the EQ pages are marked dirty to make sure they are transferred in the migration sequence.

  3. Capture the state of the source targeting, the EQs configuration and the state of thread interrupt context registers.

  Restore is similar:

  1. Restore the EQ configuration. As targeting depends on it.
  2. Restore targeting
  3. Restore the thread interrupt contexts
  4. Restore the source states
  5. Let the vCPU run