System z PCIe architecture virtualization implementation on Linux using KVM and QEMU

An introduction to zPCI virtulization

Yi Min Zhao / zyimin@cn.ibm.com

Agenda

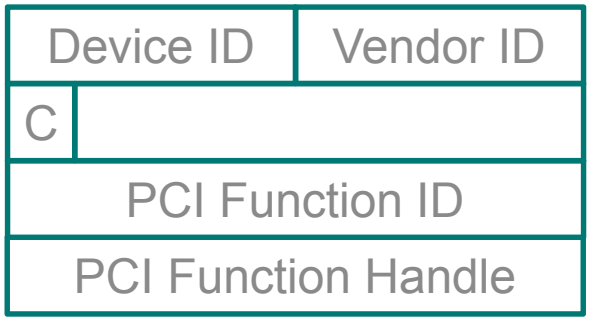
- weird PCI on System Z
- zPCI on Linux
- zPCI virtualization in KVM/QEMU
- next stage

weird PCI on System Z

- PCI is a relative newcomer to the System Z
- Only certain cards supported (RoCE, Flash, Compression)
- No MMIO
- Various instructions for reading/writing memory
- Integration into existing I/O infrastructure (adapter interrupts, channel-subsystem machine checks), only MSI-X
- No IOMMU
- No topology

zPCI on Linux – Scan PCI devices

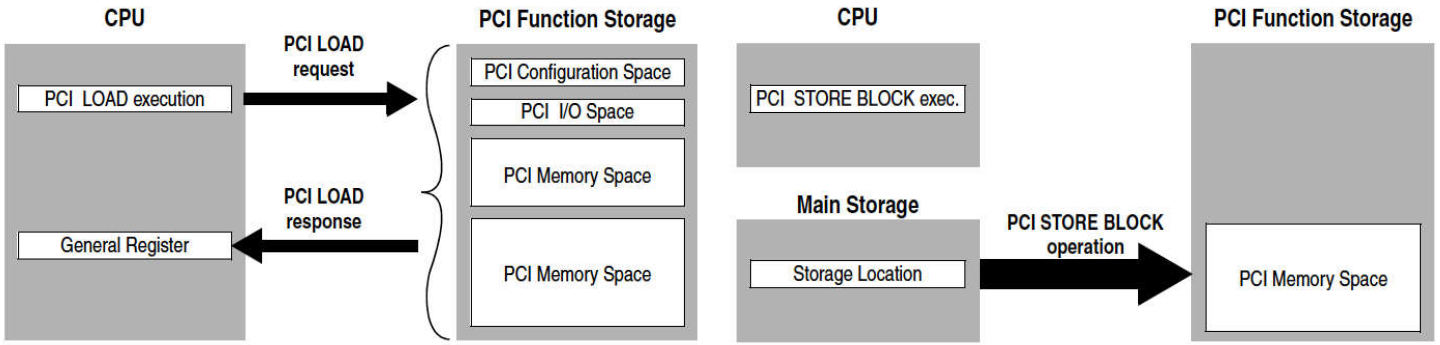
- CLP instruction – List PCI functions
  - Device ID, Vendor ID, FID, FH
- CLP instruction – Query PCI function
  - BARs, DMA values, UID
- No bus/slot/function topology
- FID & FH unstable, UID stable



domain	bus	slot	function
UID :	0000 :	00 :	0

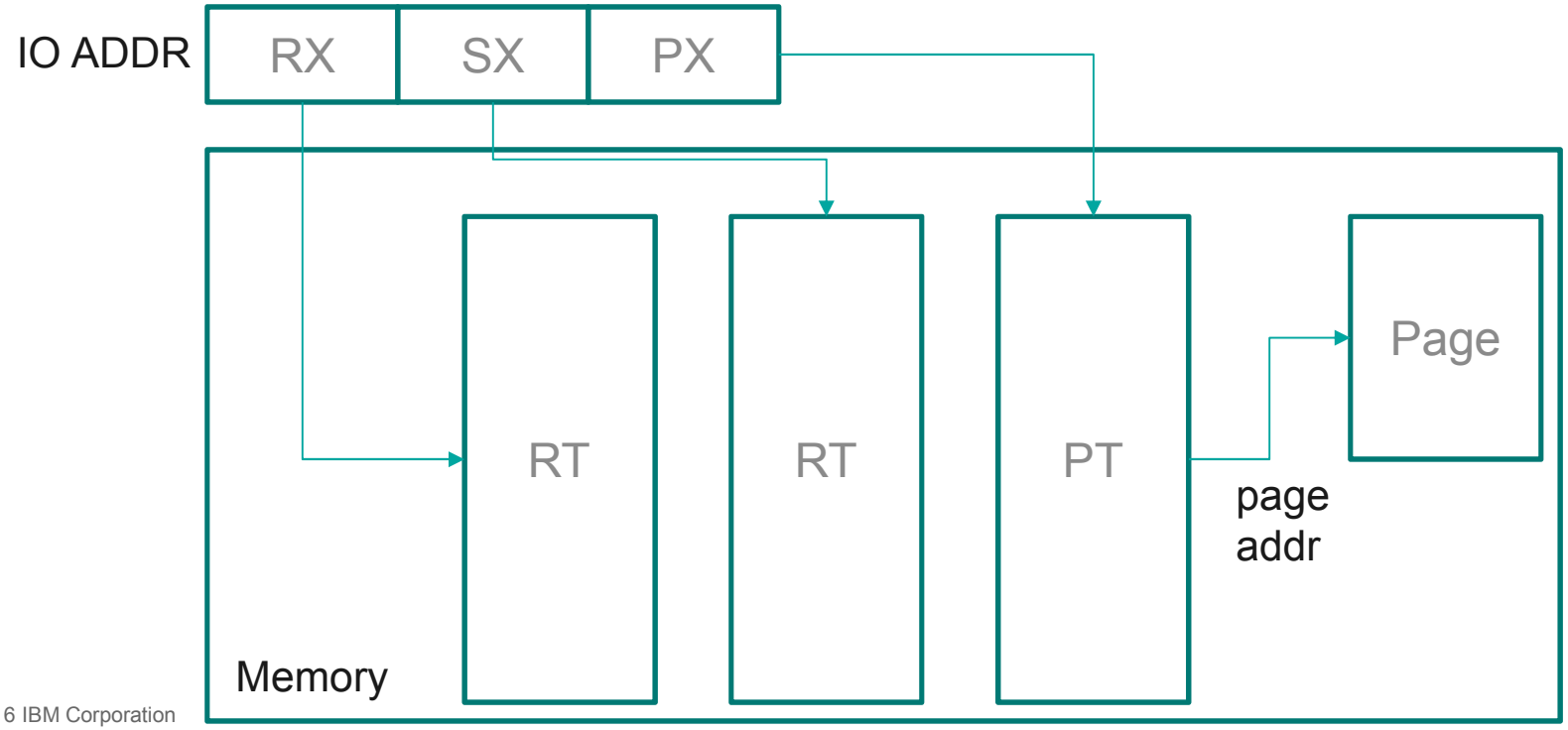
zPCI on Linux – R/W PCI device

- Read/Write PCI config space
  - pcilg/pcistg/pcistb privileged instructions



zPCI on Linux – DMA

- rpcit privileged instruction
- IO ADDR → Memory ADDR



zPCI on Linux – Adapter Interruption

- Bitmaps
  - AISB (mapping devices), AIBV (mapping msi-x entries)
- Register AIRQ
  - mpcifc privileged instruction
- Interrupt Suppression
  - scan AISB & AIBV bitmaps
  - sic privileged instruction

zPCI virtualization in KVM/QEMU - Interception

- exit SIE to KVM
- KVM to QEMU
- zpci privileged instruction interception
- re-enter SIE

zPCI virtualization in KVM/QEMU – IOMMU

- one-zpci-per-iommu
- intercept rpci privileged instruction
- walk guest DMA routing table
  - get the guest's page addr mapped by IO addr
- notify the listener of iommu memory region

zPCI virtualization in KVM/QEMU – AIRQ

- floating interrupt
  - flic qdev
- set the corresponding bits of guest's AISB and AIBV
- suppress irq injection
  - return to normal mode after intercept guest SIC instruction
- inject AIRQ
  - kvm_flic (ioctl), qemu_flic

zPCI virtualization in KVM/QEMU – Hot(un)plug

- SCLP event notification

zPCI virtualization in KVM/QEMU – Modelling

- one-pci-per-zpci
  - zpci qdev, uid/fid/target properties
- example:

```
-zpci uid=8,fid=2,target=vfio1 \
-vfio-pci host=0001:00:00.0,id=vfio1

-zpci target=vfio2 \
-vfio-pci host=0001:00:00.0,id=vfio2

-virtio-blk-pci id=virtio1

-virtio-net-pci
```

next stage

- performance
  - pcilg/stg interpertion in SIE
  - airq injection interpretation
- functionalities
  - function measurement block
  - multifunction support

Thanks!