KVM on Power

Zhang Li
2014.10.19

Zhang Li

Staff software engineer in IBM LTC. Focus on KVM development, currently work on Power. Commit Libvirt, QEMU for Power patches to community. Start to look at kernel recently. Familiar with KVM on Power architecture and system management stack.

If any questions, please send mail to zhlcindy@gmail.com

Outline

- Overview
  - Why KVM on Power
  - Power7, Power8 support
  - Cloud solutions (ovirt, openstack, docker, kimichi, ginger...)
- Power architecture
  - CPU
  - MMU
  - IO
- KVM on Power
  - Architecture of KVM on Power
  - CPU && Memory virtualization
  - Difference between Power and X86
  - IBM's contribution to KVM on Power

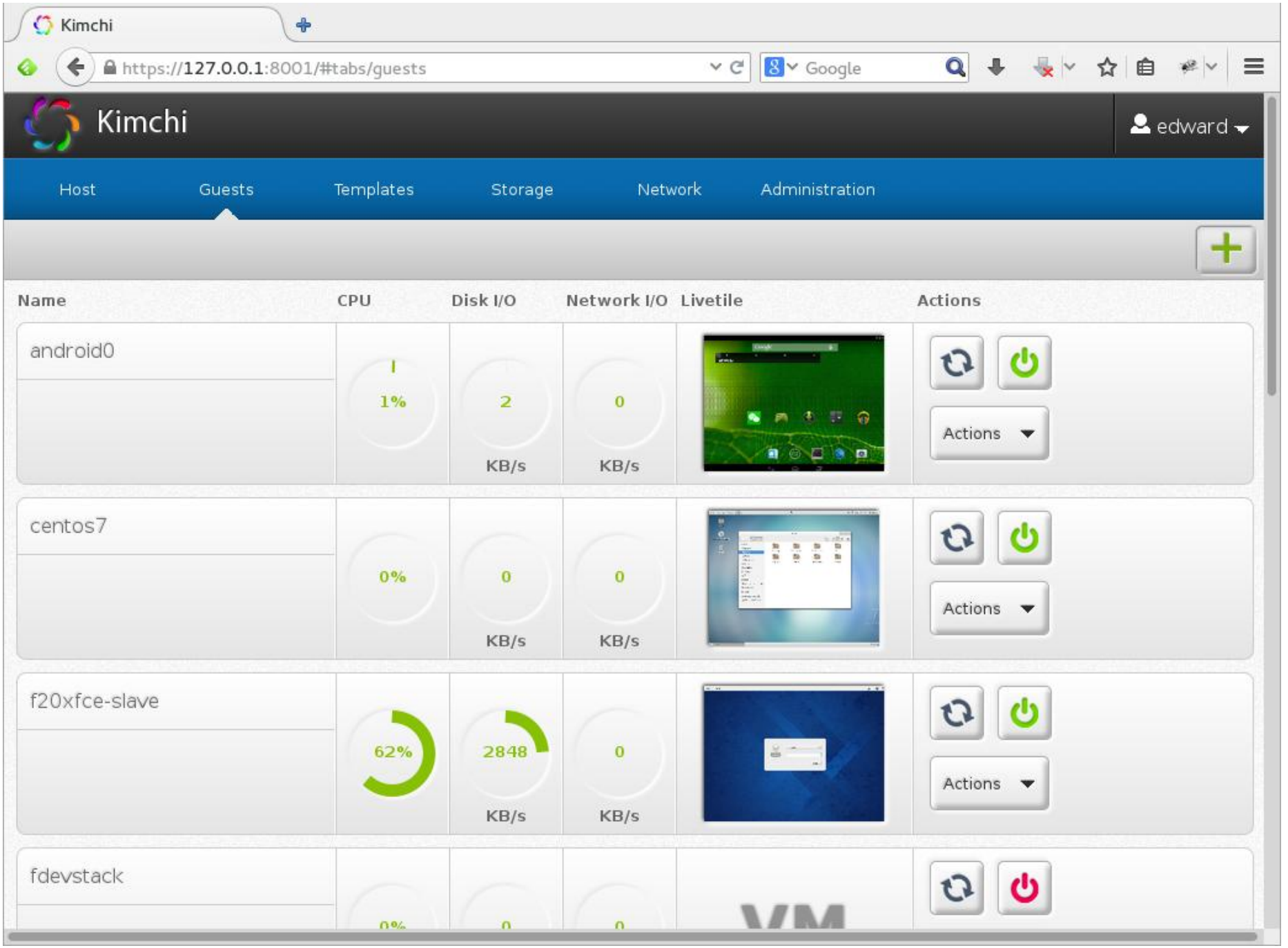
Why KVM on Power

- Power virtualization (PowerVM) has long history
- KVM becomes more popular than ever
- It's easy to support KVM
- Collaborative innovation with community
- Power is open

Cloud solutions

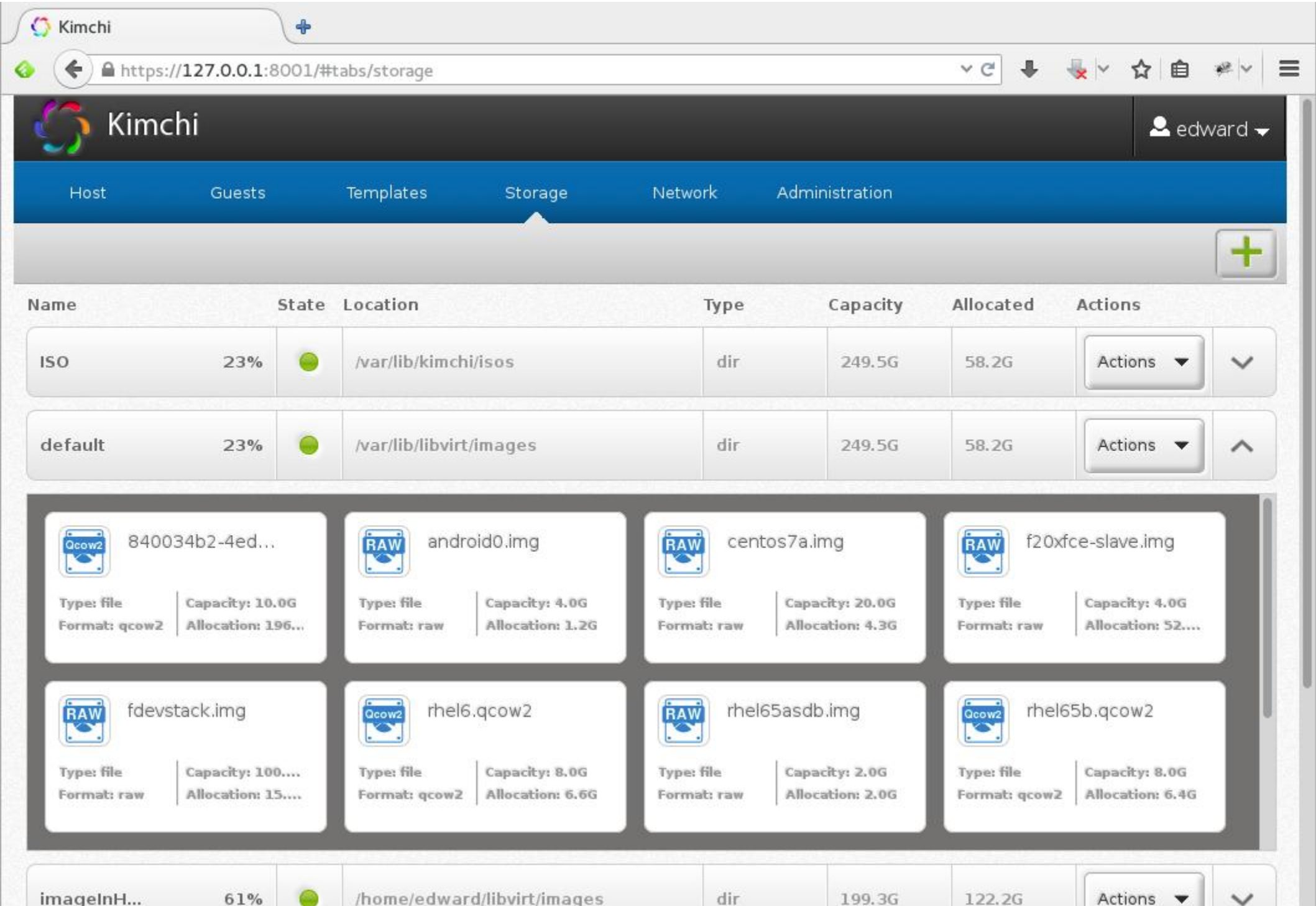
- Ovirt
- Openstack
- Docker
- Kimichi/Ginger

kimichi



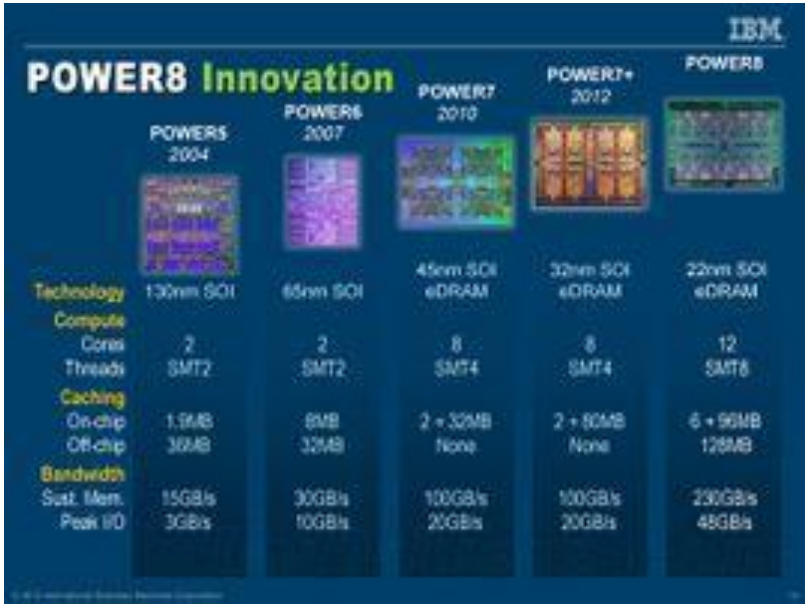
kimichi

IBM


Kimchi
https://127.0.0.1:8001/#tabs/storage
Kimchi
edward
Host
Guests
Templates
Storage
Network
Administration
Name
State
Location
Type
Capacity
Allocated
Actions
ISO 23% /var/lib/kimchi/isos dir 249.5G 58.2G Actions
default 23% /var/lib/libvirt/images dir 249.5G 58.2G Actions
840034b2-4ed...
Type: file
Format: qcow2
Capacity: 10.0G
Allocation: 196...
android0.img
Type: file
Format: raw
Capacity: 4.0G
Allocation: 1.2G
centos7a.img
Type: file
Format: raw
Capacity: 20.0G
Allocation: 4.3G
f20xfce-slave.img
Type: file
Format: raw
Capacity: 4.0G
Allocation: 52....
fdevstack.img
Type: file
Format: raw
Capacity: 100....
Allocation: 15....
rhel6.qcow2
Type: file
Format: qcow2
Capacity: 8.0G
Allocation: 6.6G
rhel65asdb.img
Type: file
Format: raw
Capacity: 2.0G
Allocation: 2.0G
rhel65b.qcow2
Type: file
Format: qcow2
Capacity: 8.0G
Allocation: 6.4G
imageInH... 61% /home/edward/libvirt/images dir 199.3G 122.2G Actions

Power CPU support

- BOOK3s: Power7, Power8, 970…
- BOOK3E



Outline

- Overview
  - Why KVM on Power
  - Power7, Power8 support
  - Cloud solutions (ovirt, openstack, docker, kimichi, ginger...)
- Power architecture
  - CPU
  - MMU
  - IO
- KVM on Power
  - Architecture of KVM on Power
  - CPU && Memory virtualization
  - Difference between Power and X86
  - IBM's contribution to KVM on Power

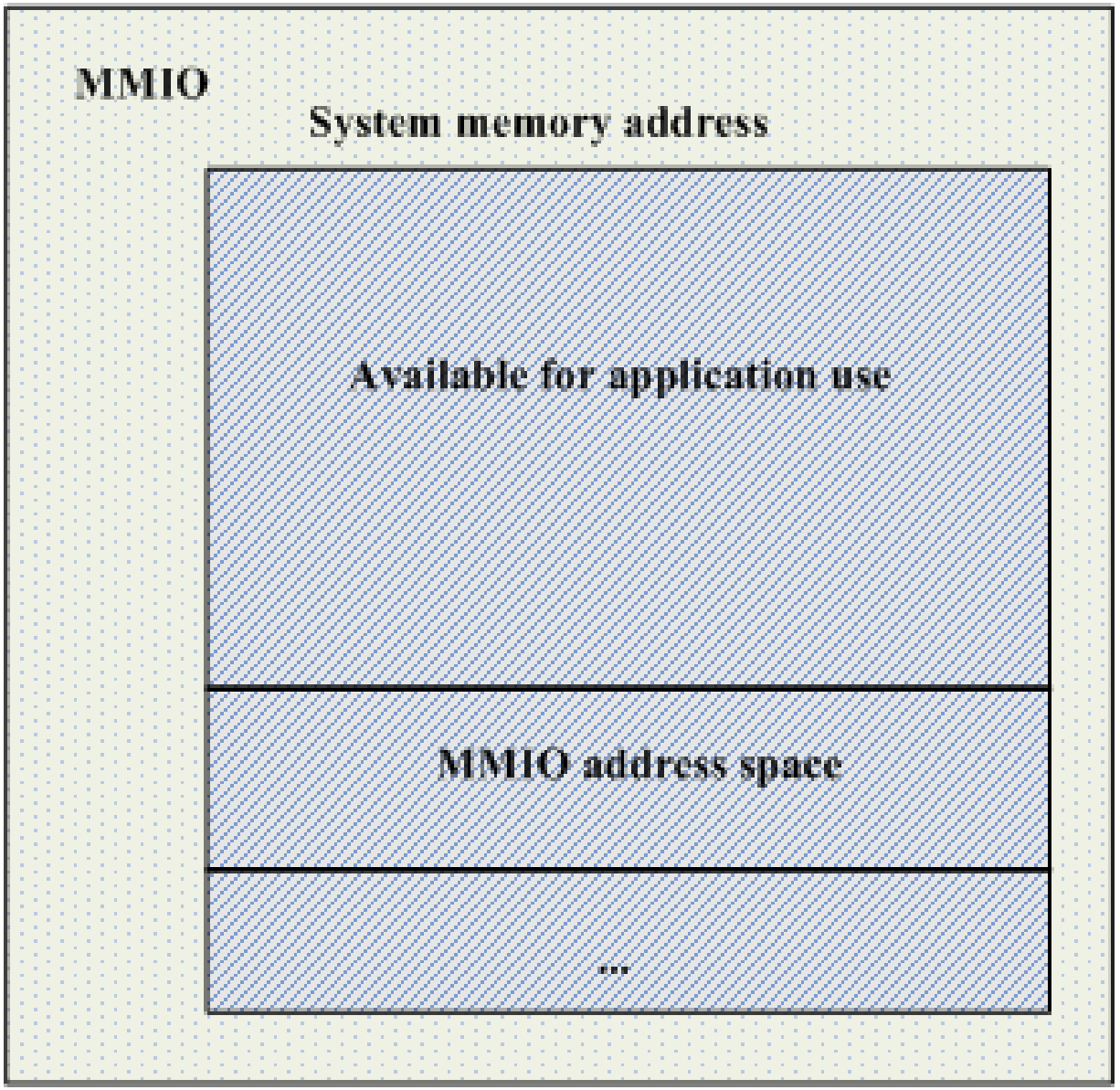
Power (BOOK3s) Architecture

- CPU instruction set, compliant with PowerISA. 2.0.7 is last revision. Available from www.power.org
- Memory management specified by PowerISA
- I/O and PCI architecture defined by PAPR (Power Architecture Platform Requirements)

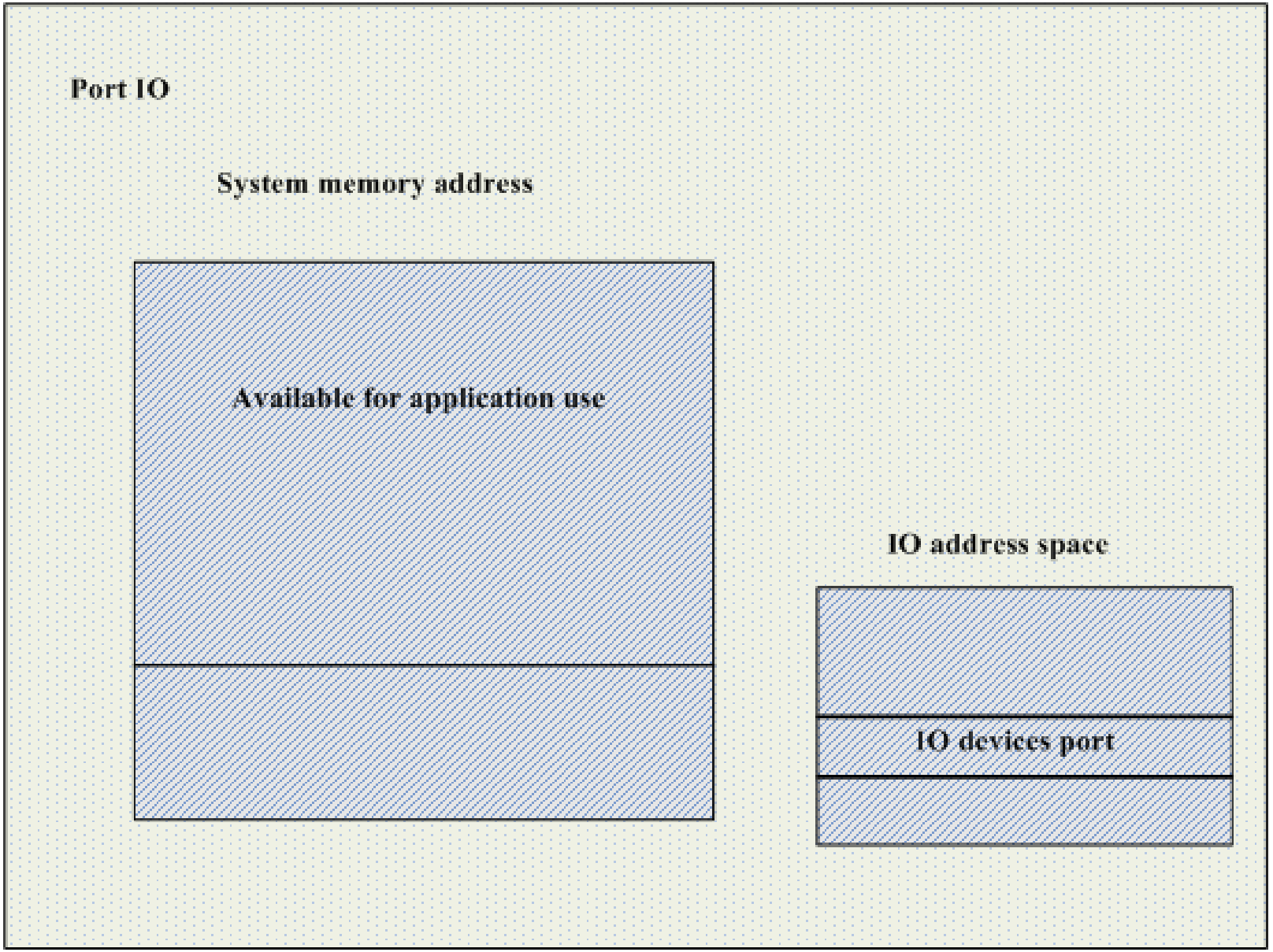
CPU

- RISC instructions set
  - Many instructions; fixed-length instruction(128-bits)
  - Hypervisor/privileged/problem state
- CPU is big-endian
- Fairly support partition/virtualization
- Simultaneous Multithreading (SMT)
- All IO is memory-mapped

MMIO on Power



Port IO on X86



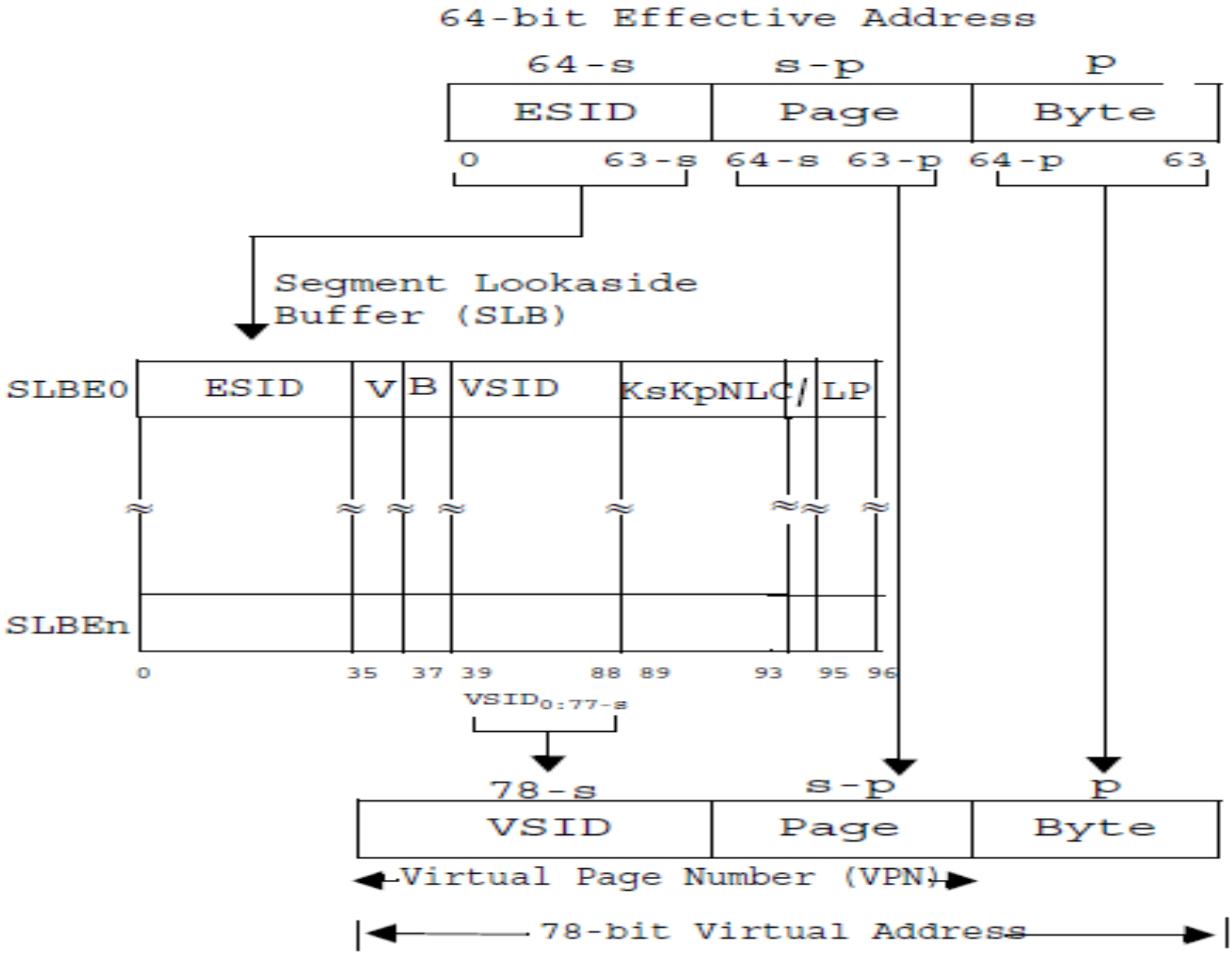
MMU

- Effective/virtual/physical address
- Segment Lookaside Buffer (SLB)
- Hashed Page Table (HTAB)

MMU- segmenting

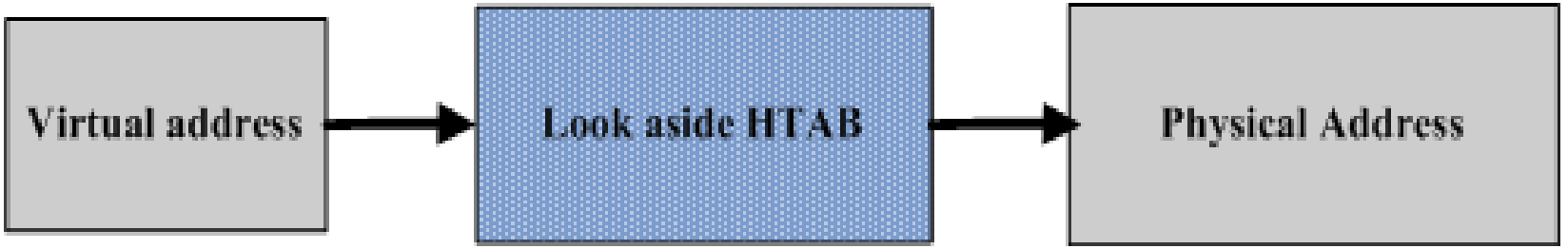
- Segment Lookaside Buffer (SLB)
  - Translate ESID to VSID (Segment ID)
  - Translate 64-bits effective address to 78-bits virtual address

Virtual Address generation



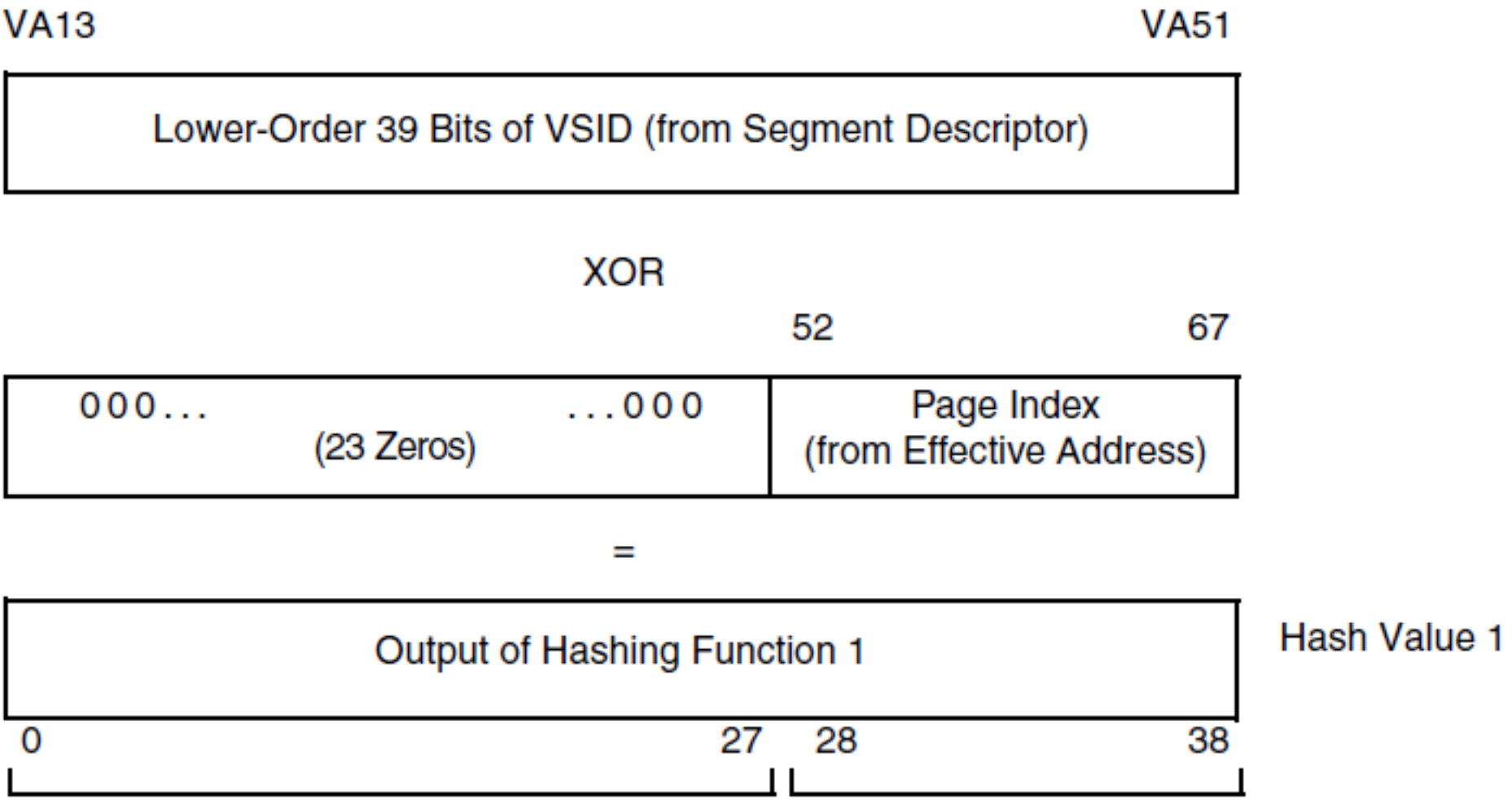
MMU - Paging

- Hashed Page Table (HTAB)
- Translate 78-bits virtual address to 60-bits physical address



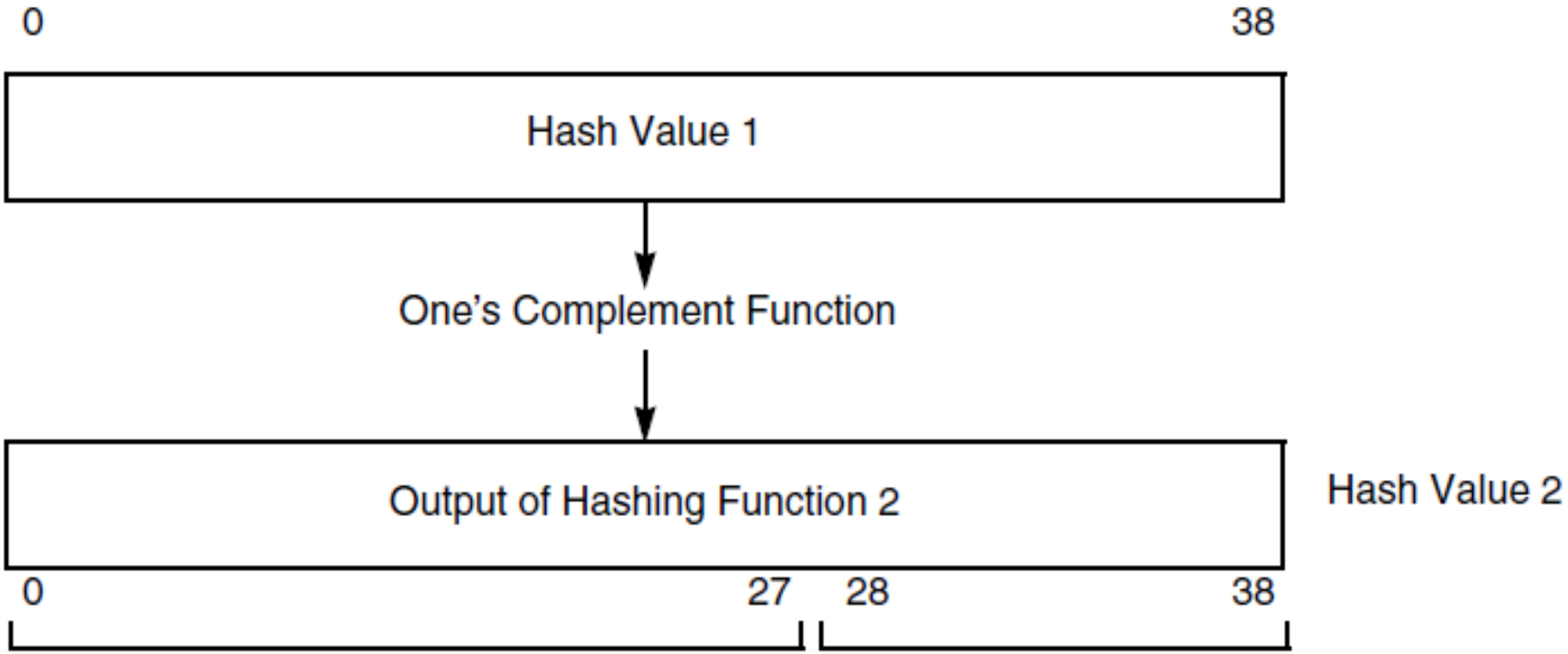
MMU-Paging

Primary Hash:



MMU-Paging

Secondary Hash:

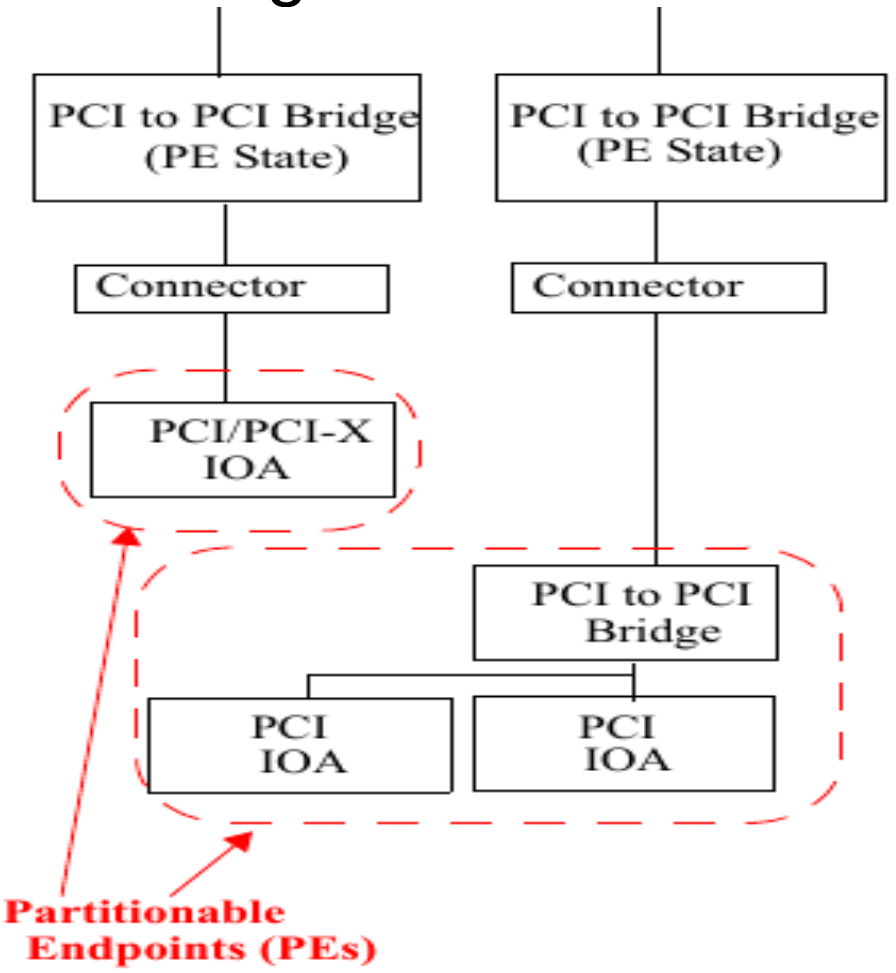


IO (PCI)

- Compatible with IODA1/2
- Compatible with PCI GEN2/3 spec
- PEs (Partitionable Endpoints)
- IOAs (IO Adapters)

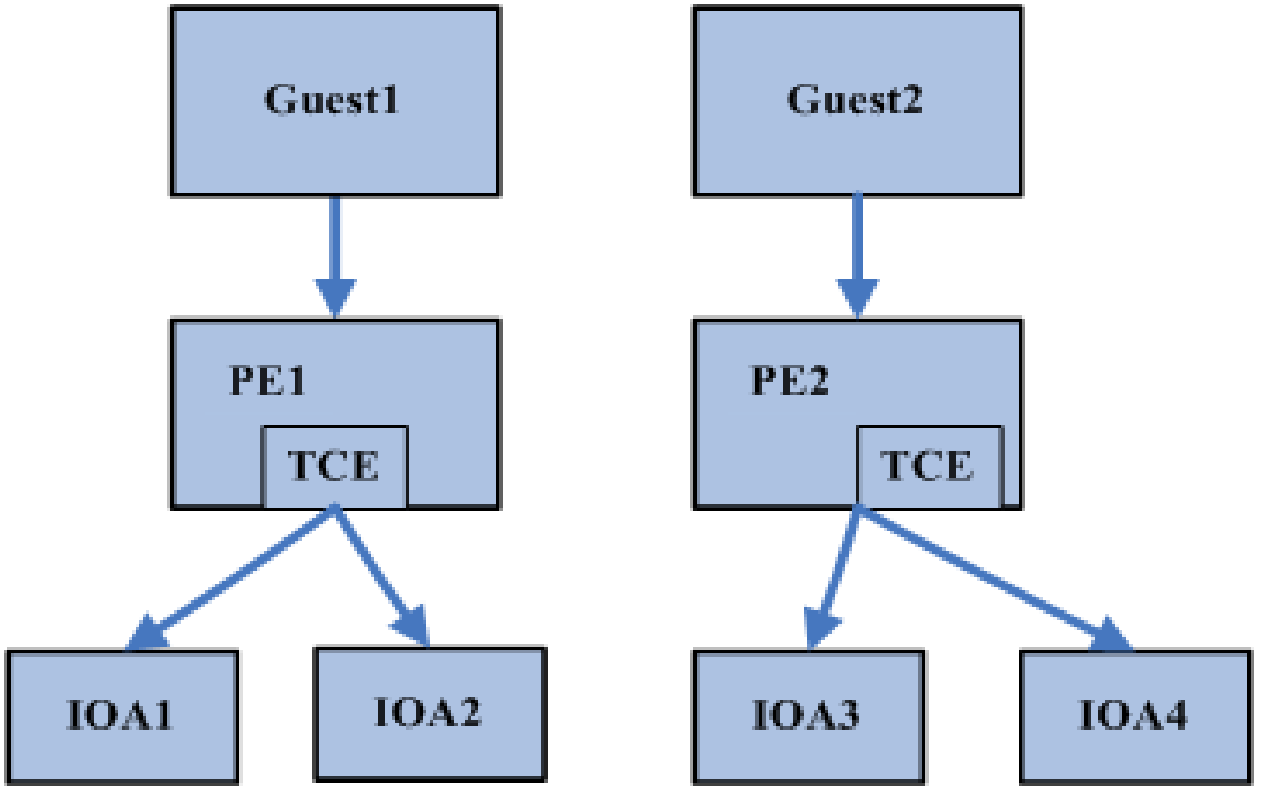
PCI-PE

- Isolated domain for individual power, error, and management etc.
- Helps to partitioning PCI domain

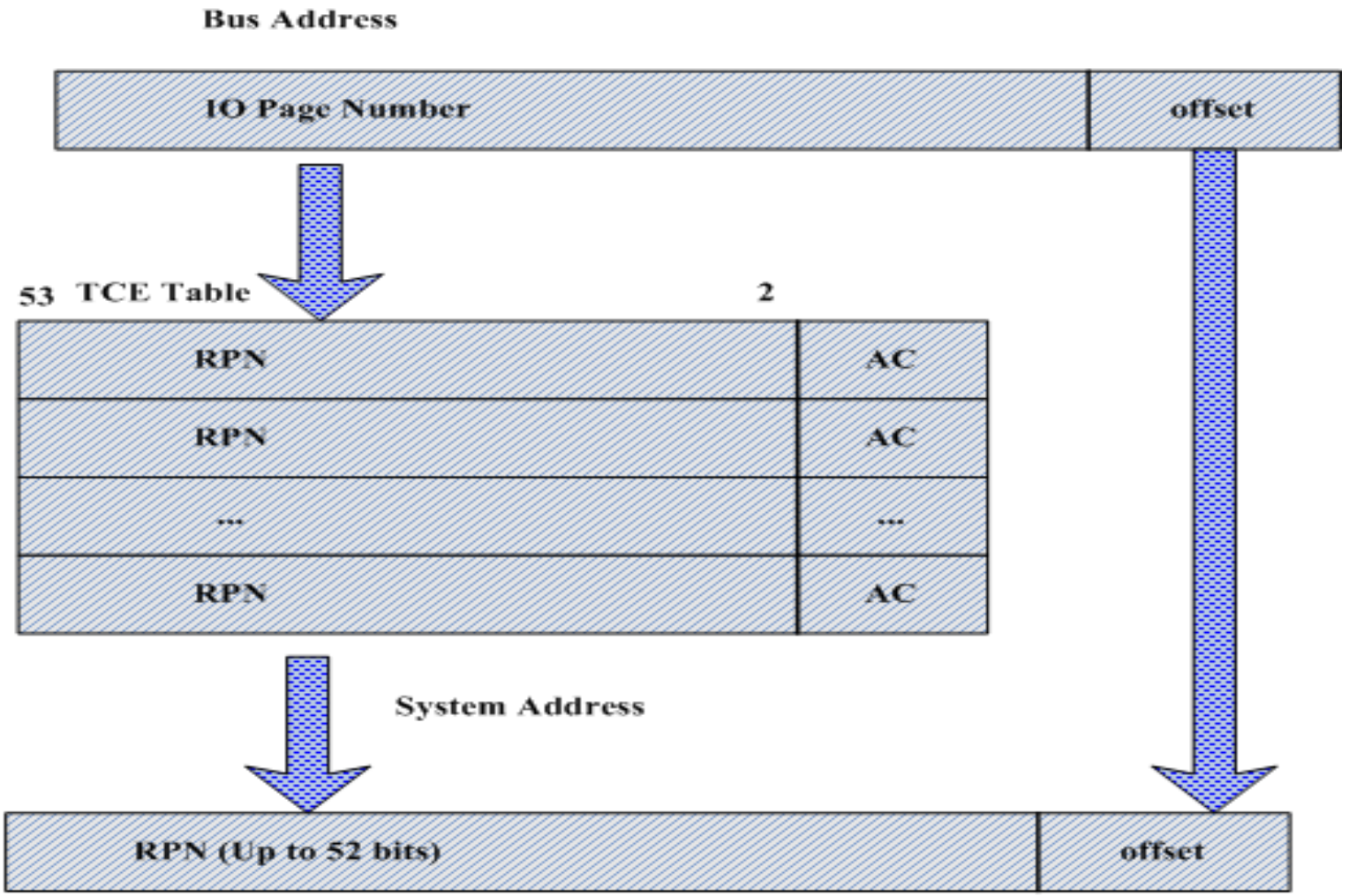


IOMMU/TCE Table

- Translation Address (PCI bus address -> memory)
- Ability to support (read-write) protection and migration
- Each PE has individual TCE table



TCE

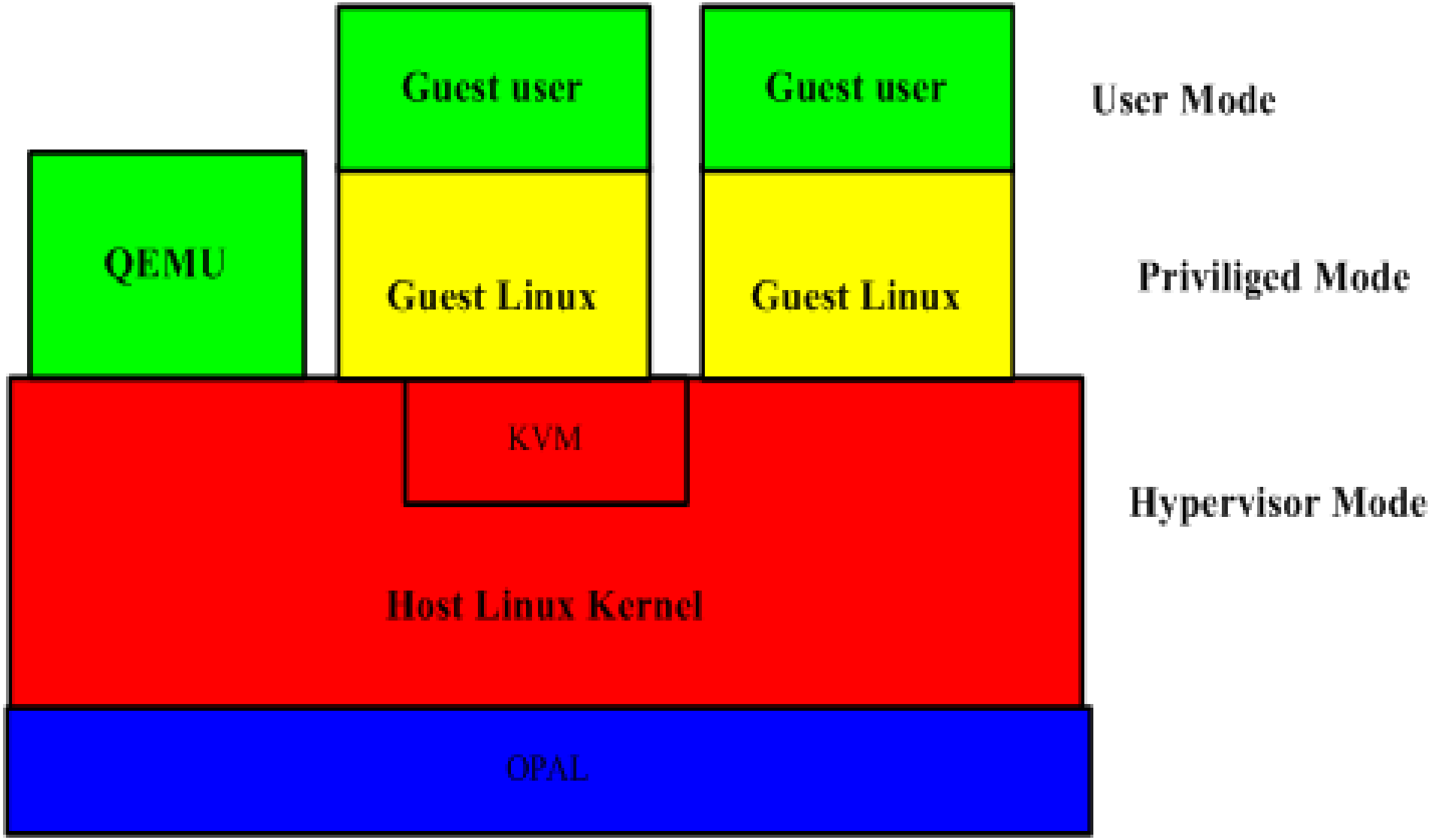


Access Control (2 bits)
00 page fault (no access)
01 System address sapce (read only)
10 System address space (write only)
11 System address space (read/Write)

Outline

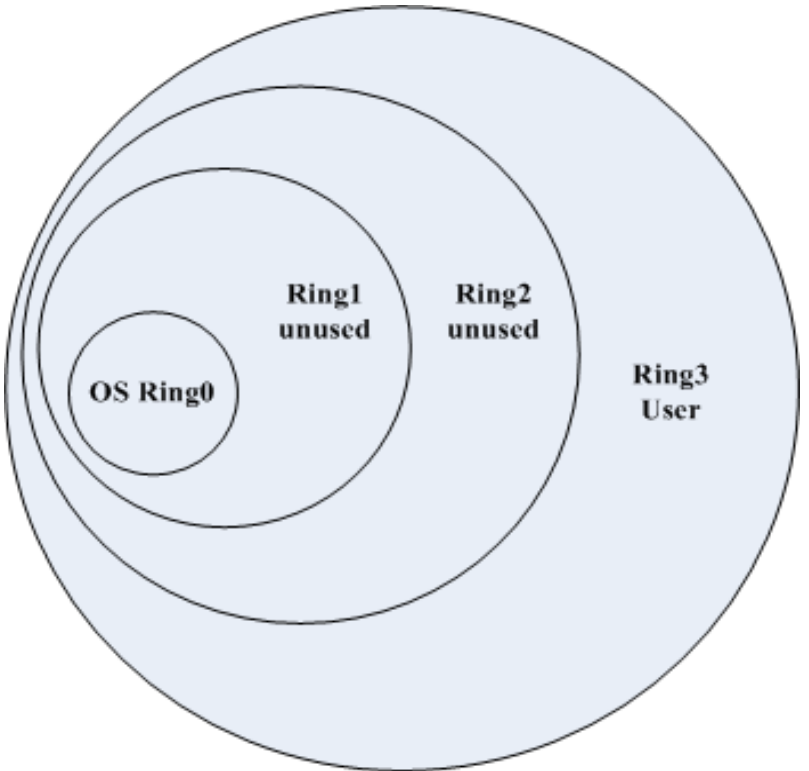
- Overview
  - Why KVM on Power
  - Power7, Power8 support
  - Cloud solutions (ovirt, openstack, docker, kimichi, ginger...)
- Power architecture
  - CPU
  - MMU
  - IO
- KVM on Power
  - Architecture of KVM on Power
  - CPU & memory virtualization
  - Difference between Power and X86
  - IBM's contribution to KVM on Power

Architecture of KVM on Power



CPU Modes

CPU Mode on Power	CPU mode on X86
Hypervisor Mode	Ring0
Privileged Mode	Ring1
User Mode	Ring3

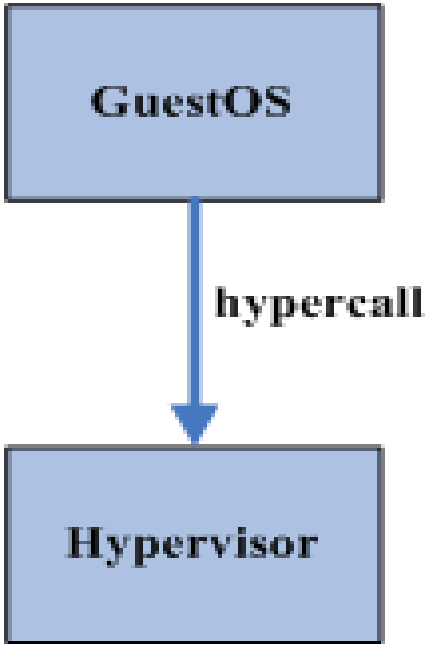


Hypervisor mode

- Control MMU hash table and devices access
- Control of which interrupts go to the guest directly and which one go to the hypervisor
- Run some instructions and access special-purpose registers(SPRs)

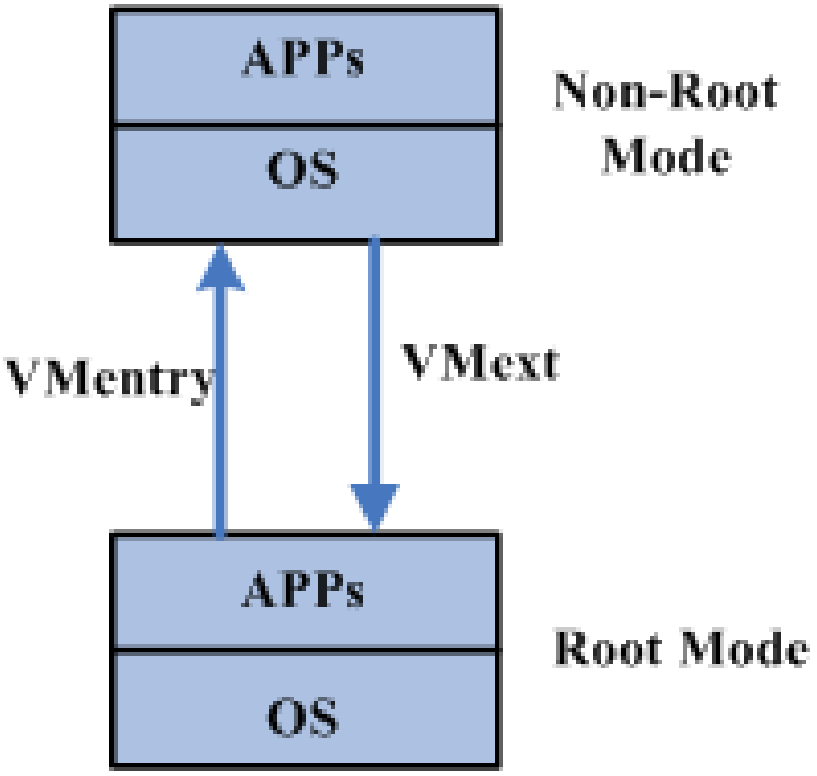
CPU Modes Switch

- Hypercall
  - Instruction: sc 1
- Exception
  - HDSI/HISI etc.
- External interrupts
  - Even decrementer



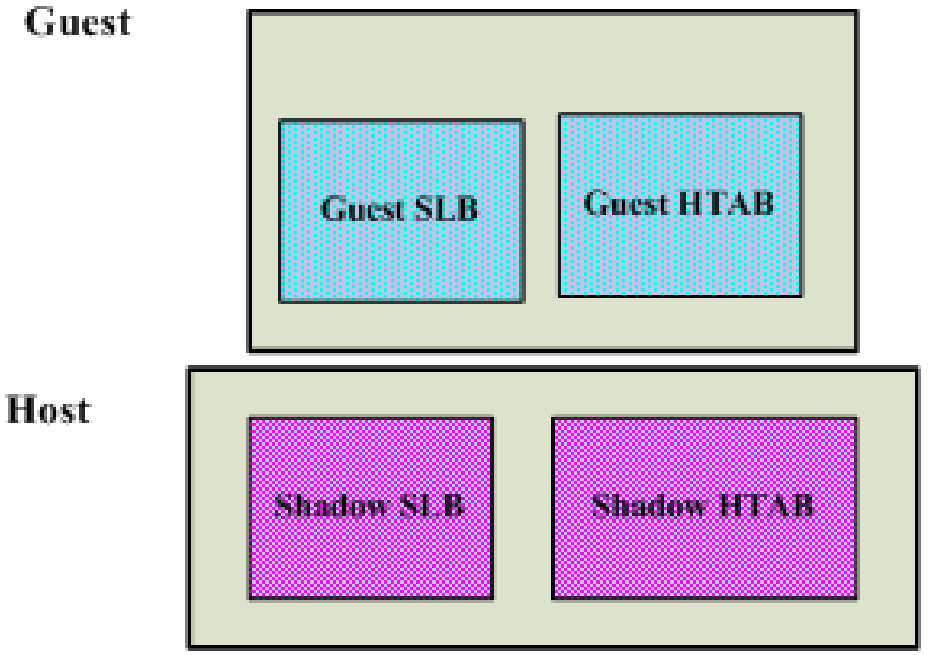
CPU Levels on X86

- Root Mode (ring 0, ring3)
- Non-Root mode (ring0, ring3)



Memory virtualization

- Shadow SLB and page table maintained by host
- Propagate page fault of guest to host
  - Shadow SLB/HTAB updated
  - Propagate to guest, which updates shadow SLB/HTAB via hypercall



Para-virtualization design on Power

- RTAS Call (Run Time Abastraction Service)
  - Specified in PAPR
  - Channel for host/guest to exchange information
  - Implemented with help of dedicated hypercall
- PAPR(Power Architecture Platform Requirement)
  - Virtualization interfaces are defined
  - Hypercall is provided to virtualization
  - Guests kernel are not necessary to modified, PAPR can provide the interface.

Full-Virtualization design on X86

- Hardware virtualization
  - VMX, SVM,
  - VT-D
- System levels
  - Root Mode
  - Non-Root Mode

IBM's Contribution to community

- KVM on Power
  - Virtio
  - SRIOV
  - PCI passthrough
  - VFIO
  - Hotplug
- OpenFirmware
  - OPAL/SLOF

IBM's Contribution to community

- Linux on Power
  - https://git.kernel.org/
- KVM on Power
  - Codes are merged to upstream
- QEMU for Power
  - git://git.qemu.org/qemu.git
- Latest Linux version(3.18.rc1) can support Linux & KVM pretty well

Conclusion

- Overview
  - Power is open
  - Power virtualization solutions are all supported
- Power Architecture
  - PowerISA/PAPR define architecture, Refer: www.power.org
- KVM on Power
  - It’s para-virtualization!!! Different from X86.
  - Latest Linux kernel can support

Q&A

Thanks. ☺