SNIA DEVELOPER CONFERENCE

SDC 24

BY Developers FOR Developers

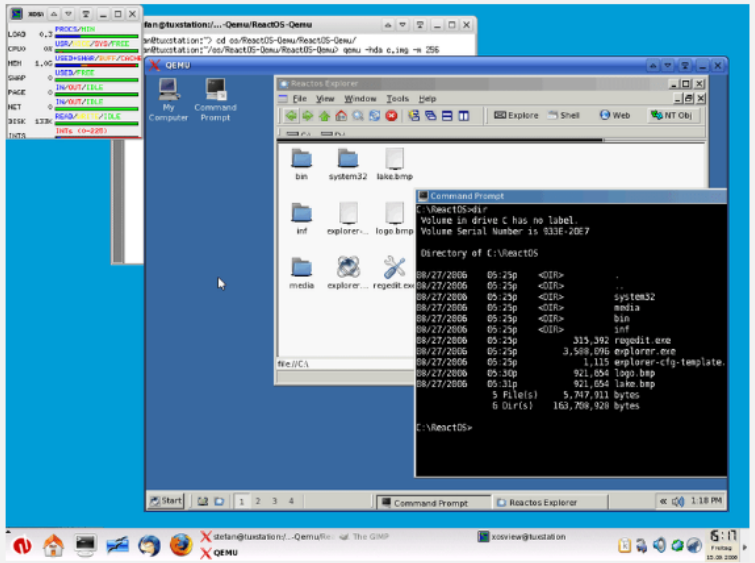
September 16-18, 2024
Santa Clara, CA

Emulating CXL with QEMU

Adam Manzanares
Samsung

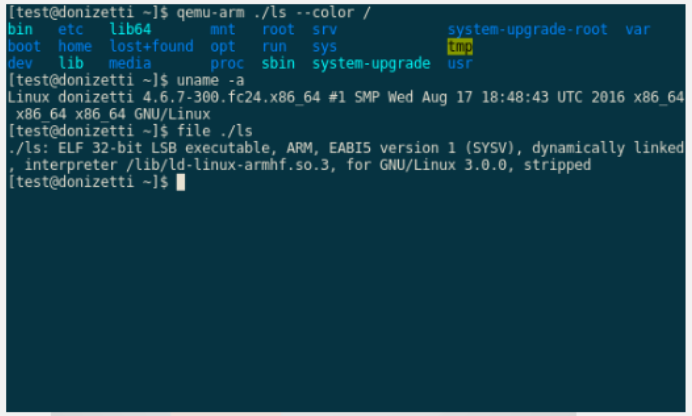
What is QEMU

- Generic and open source machine emulator and virtualizer[1]



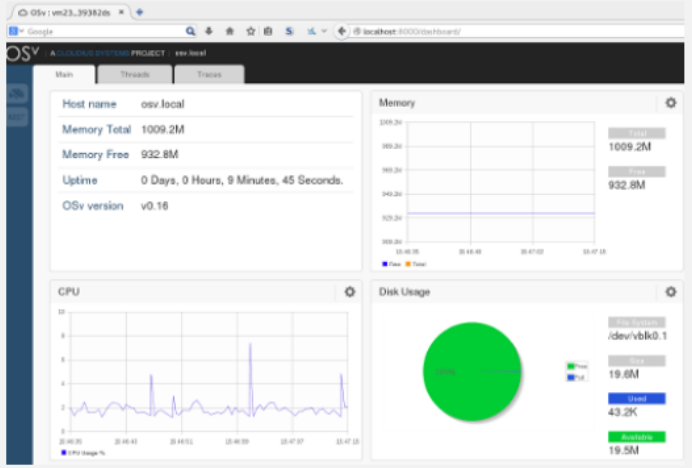
Full-system emulation

Run operating systems for any machine, on any supported architecture



User-mode emulation

Run programs for another Linux/BSD target, on any supported architecture



Virtualization

Run KVM and Xen virtual machines with near native performance

[1] https://www.qemu.org/

QEMU for NVME Emulation

- Great tool for developers
  - Ability to rapidly prototype end-to-end SW for new features
    - ZNS, FDP, simple copy, SR-IOV
    - Create host software that leverages these features
      - Complete plumbing
- Samsung Successes
  - NVMe Support
    - Maintainer – Klaus Jensen, Reviewer - Jesper Devantier
  - Testing frameworks can move faster than hardware availability
  - Brings more people into ecosystem

Why CXL Device Emulation for CXL

- Reproduce success cases for NVMe
  - Build end-to-end SW without HW dependence
- CXL Limited HW Availability
  - Emerging technology
- Benefits two ecosystems
  - OS
  - BMC

CXL System Components Emulated

- CXL Fixed Memory Windows (CFMW)
  - Maps host physical address space to at least one CXL host bridge
    - Interleave and quality of service throttling handled here
- CXL Host Bridge
  - Similar to PCIe host bridge
  - Has HDM (Host defined memory decoder)
    - Maps to root ports under host bridge
- CXL Switch
  - Single upstream port, internal PCI bus, multiple downstream ports
- CXL Memory Devices (Type 3)
  - Volatile and Persistent Regions

Example Topologies[1]

Fig 1

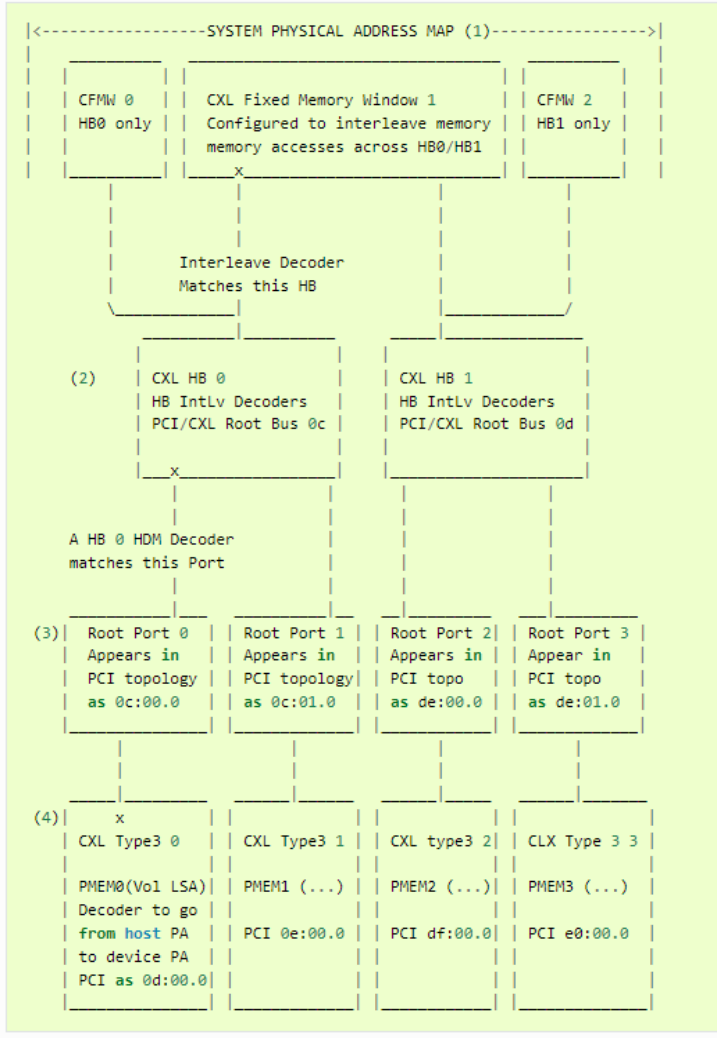
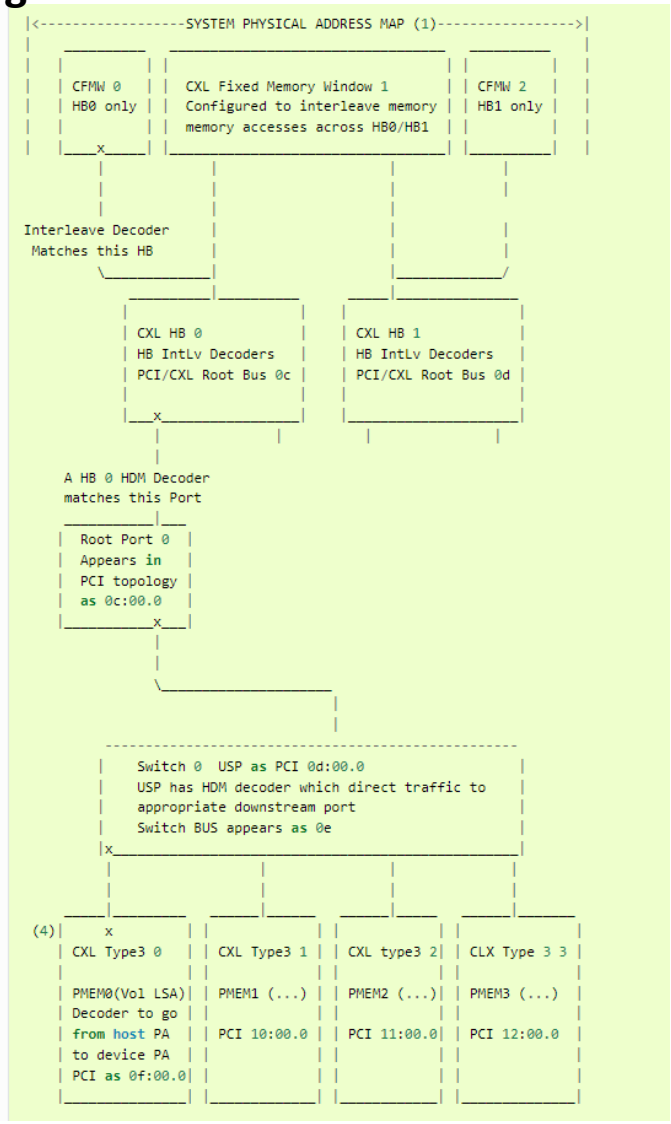


Fig 2



- Fig 1
  - 3 Fixed Windows
    - 1 Active
  - Two Host Bridges
    - Each with two root ports
  - 4 Type 3 Persistent Devices
    - Connected to root ports
- Fig 2
  - Host bridge up
    - Same as Fig 1
  - Root Port
    - Connect to switch
      - 4 Down stream Ports (DSP)
  - 4 Type 3 Persistent Device
    - Connected to Switch DSPs

[1] https://www.qemu.org/

It works

```
~/c/cxl-test-tool
fan@DT ~/c/cxl-test-tool (main)> cxl-tool --run -A kvm --create-topo
warning: image_name is not given with --image option, use QEMU_IMG (
/home/fan/cxl/images/qemu-image.img)
Using cxl topology created from xml ...
***: Start running Qemu...
QEMU instance is up, access it: ssh root@localhost -p 2024
fan@DT ~/c/cxl-test-tool (main)> cxl-tool --load-drv

** Task: install cxl modules **

Loading cxl drivers: modprobe -a cxl_acpi cxl_core cxl_pci cxl_port
cxl_mem cxl_pmem
Loading nd_pmem for creating region for cxl pmem
Loading dax related drivers

Module                  Size  Used by
device_dax             20480  0
nd_pmem                24576  0
nd_btt                 28672  1 nd_pmem
dax                    57344  2 device_dax,nd_pmem
cxl_pmem               24576  0
libnvdimm             221184  3 cxl_pmem,nd_btt,nd_pmem
cxl_mem                16384  0
cxl_port               16384  0
cxl_pci                28672  0
cxl_acpi               24576  0
cxl_core              323584  6 cxl_pmem,cxl_port,cxl_mem,cxl_pci,cx
l_acpi
```

```
~/c/cxl-test-tool
fan@DT ~/c/cxl-test-tool (main)> cxl-tool --cmd "cxl list -i"
Qemu: execute "cxl list -i" on VM
[
  {
    "memdev":"mem0",
    "serial":0,
    "host":"0000:0d:00.0"
  }
]
fan@DT ~/c/cxl-test-tool (main)> cxl-tool --create-dcR

** Task: Create DC region **


** Task: Show dc region **

[
  {
    "memdevs":[
      {
        "memdev":"mem0",
        "serial":"0",
        "host":"0000:0d:00.0"
      }
    ]
  },
  {
    "regions":[
      {
        "region":"region0",
        "resource":"0xa90000000",
        "size":"1024.00 MiB (1073.74 MB)",
        "interleave_ways":1,
        "interleave_granularity":256,
        "decode_state":"commit"
      }
    ]
  }
]
fan@DT ~/c/cxl-test-tool (main)>
```


Under the Hood

```
fan@leg:~/cxl/cxl-test-tool$ cxl-tool.py --create-dcR mem0
ssh root@localhost -p 2024 "cxl enable-memdev mem0"
cxl memdev: cmd_enable_memdev: enabled 1 mem
ssh root@localhost -p 2024 "echo region0 > /sys/bus/cxl/devices/decoder0.0/create_
dc_region"

ssh root@localhost -p 2024 "echo 256 > /sys/bus/cxl/devices/region0/interleave_gra
nularity"

ssh root@localhost -p 2024 "echo 1 > /sys/bus/cxl/devices/region0/interleave_ways"

ssh root@localhost -p 2024 "echo dc0 >/sys/bus/cxl/devices/decoder2.0/mode"

ssh root@localhost -p 2024 "echo 0x40000000 >/sys/bus/cxl/devices/decoder2.0/dpa_s
ize"

ssh root@localhost -p 2024 "echo 0x40000000 > /sys/bus/cxl/devices/region0/size"

ssh root@localhost -p 2024 "echo  decoder2.0 > /sys/bus/cxl/devices/region0/target
0"

ssh root@localhost -p 2024 "echo 1 > /sys/bus/cxl/devices/region0/commit"

ssh root@localhost -p 2024 "echo region0 > /sys/bus/cxl/drivers/cxl_region/bind"

ssh root@localhost -p 2024 "cxl list -r region0"
[
  {
    "region":"region0",
    "resource":45365592064,
    "size":1073741824,
    "interleave_ways":1,
    "interleave_granularity":256,
    "decode_state":"commit"
  }
]
DC region region0 created for mem0
fan@leg:~/cxl/cxl-test-tool$
```


Features You Can Emulate

- Events, FW Update, Timestamp, Logs, Identify, Sanitize, Poison Management, Get MHD info, DCD, Switch – [identify, logs, port state, tunnel management command]
- MCTP and Switch CCIs
- Visit https://gitlab.com/jic23/qemu/
  - Bleeding edge support
  - CXL branches with dates

Memory Semantic SSD Emulation

- https://github.com/SamsungDS/linux/tree/v5.18-for-msssd-qemu
- Dual Mode
  - NVMe
    - Commands and DMA based transfers
  - CXL
    - HDM range mapped onto LBA space
    - Load/store accesses
- Questions
  - Tong Zhang <t.zhang2@samsung.com>

See the following more information

- Getting started
  - https://github.com/moking/moking.github.io/wiki/Basic:-CXL-test-with-CXL-emulation-in-QEMU
- Dynamic capacity device (DCD)
  - cxl-test-tool: A tool to ease CXL test with QEMU setup--Using DCD test as an example · moking/moking.github.io Wiki · GitHub
- Chat with us
  - https://discord.gg/jmgNJywXWs

Acknowledgements

- Ben Widawsky – (Google)
- Jonathan Cameron (Huawei)
- Ira Weiny (Intel)
- Gregory Price (Meta)
- Fan Ni (Samsung)
- Davidlohr Bueso (Samsung)
- Tong Zhang (Samsung)
- Many others

Please take a moment to rate this session.

Your feedback is important to us.

