GiantVM: A Many-to-one Virtualization System Build Atop the Qemu/KVM Hypervisor

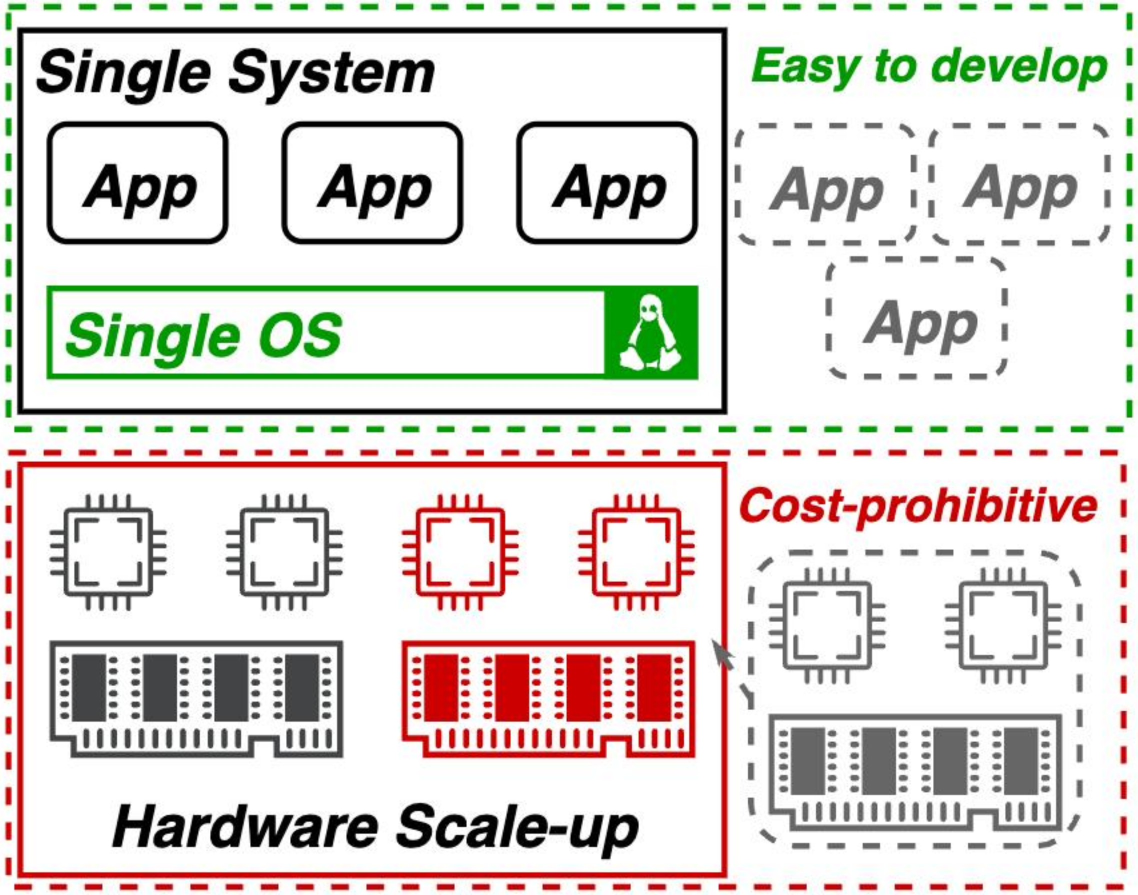
Xiong Tianlei, Xue Songtao, Muliang Shou

2025.09.05

Why do we need many-to-one virtualization?

BACKGROUND

Resource Scaling

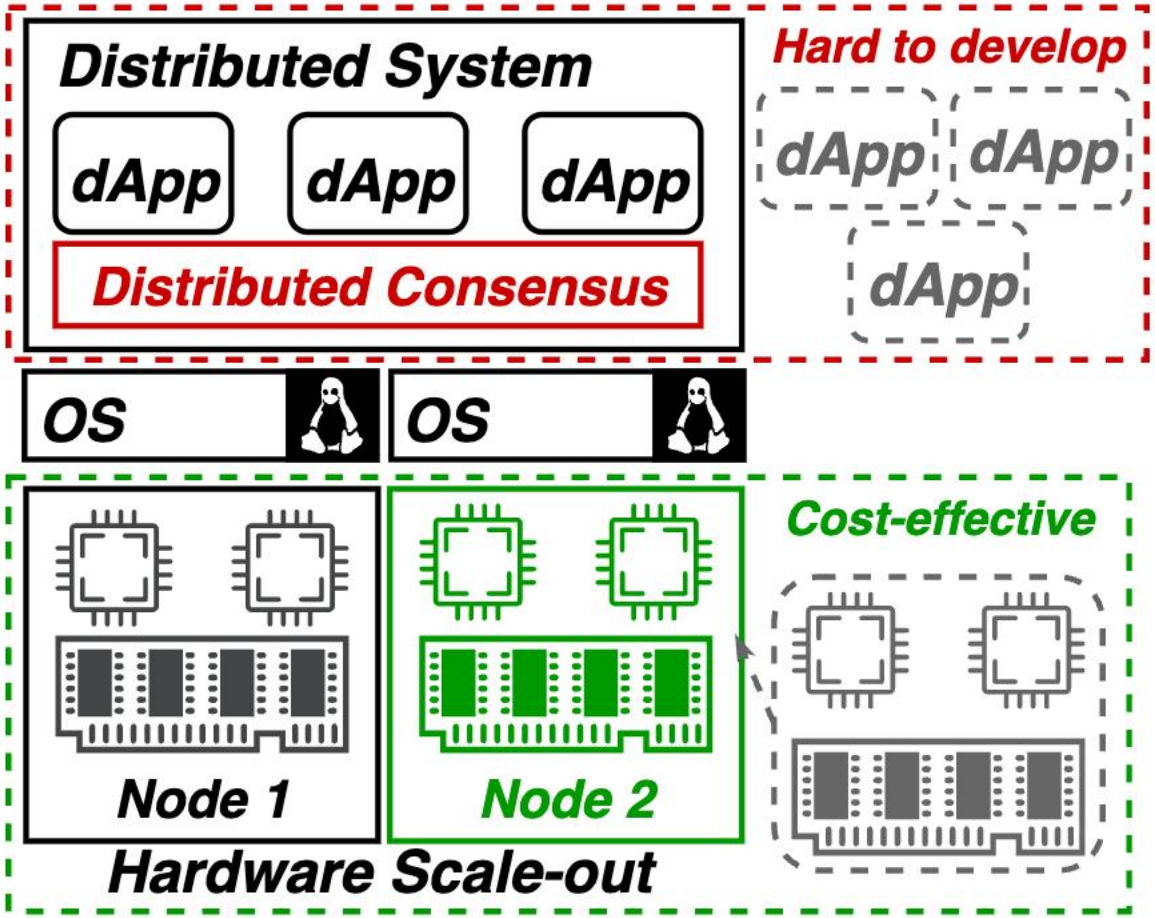


- **Scale-up**
  - Aggregate resource in one node
  - e.g. IBM zSeries

☑ **No need to port exsiting Software**

✗ **Expensive**

Resource Scaling

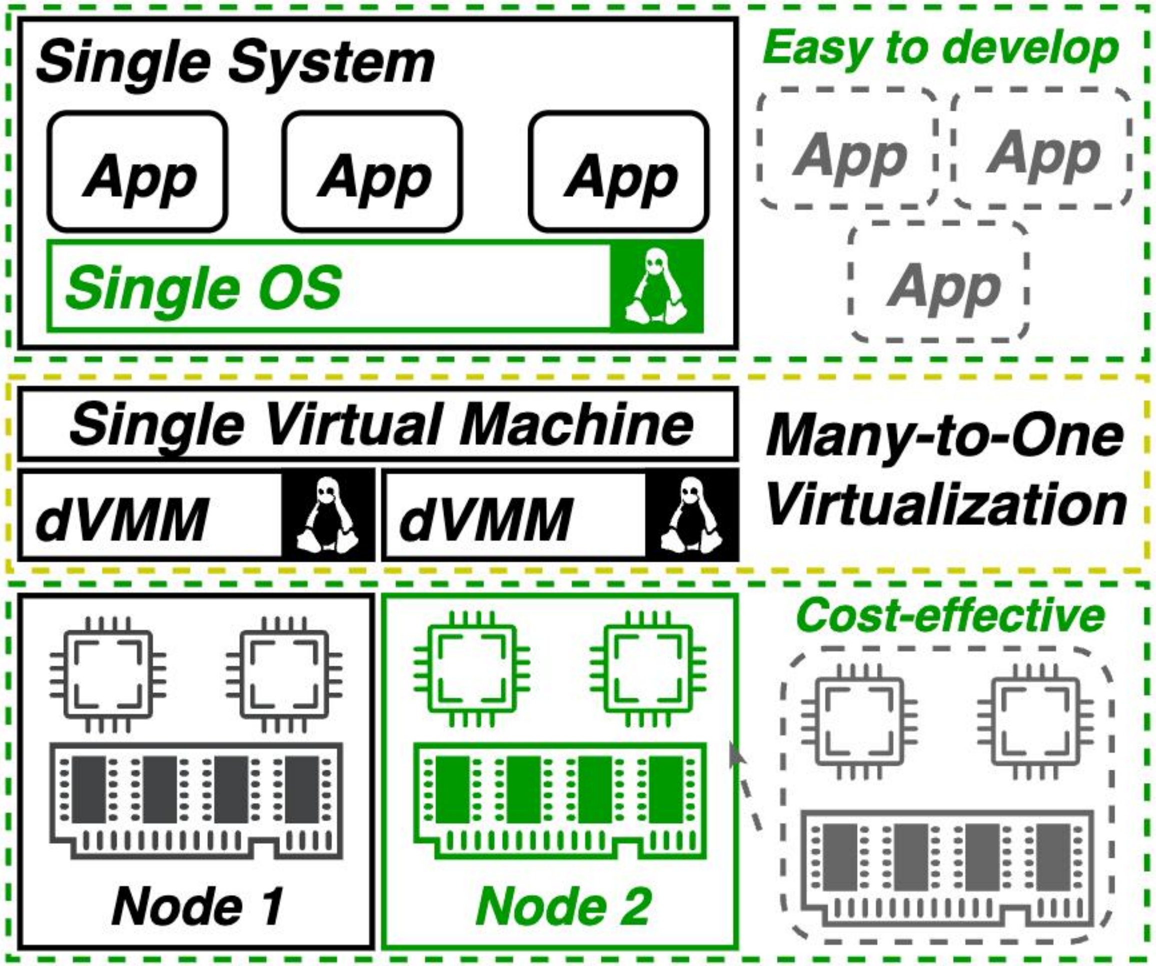


- **Scale-out**
  - Aggregate resource with more nodes
  - e.g. cluster

☑ **Affordable**

✗ **A huge engineering effort to port existing software**

Many-to-one Virtualization



- **Many-to-one Virtualization**
  - Aggregates multiple physical nodes into a single large VM

☑ **Affordable**

☑ **Single System Image**

⚠ **Performance?**

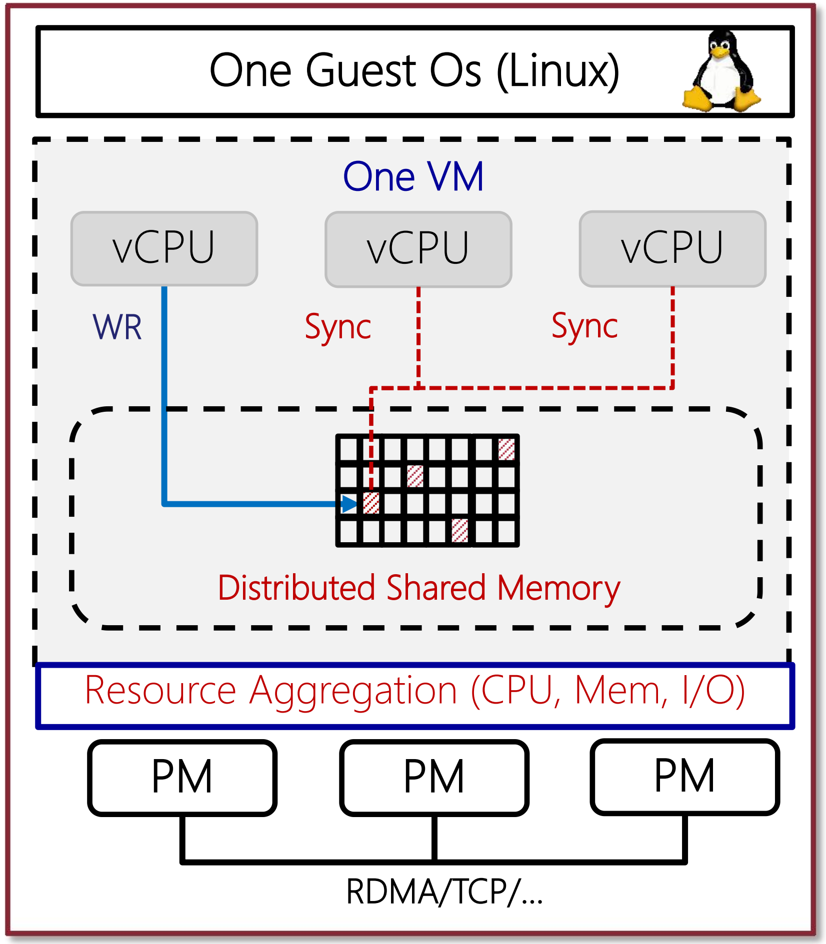
A Quick Recap of GiantVM in KVM Forum 2018

OVERVIEW OF GIANTVM

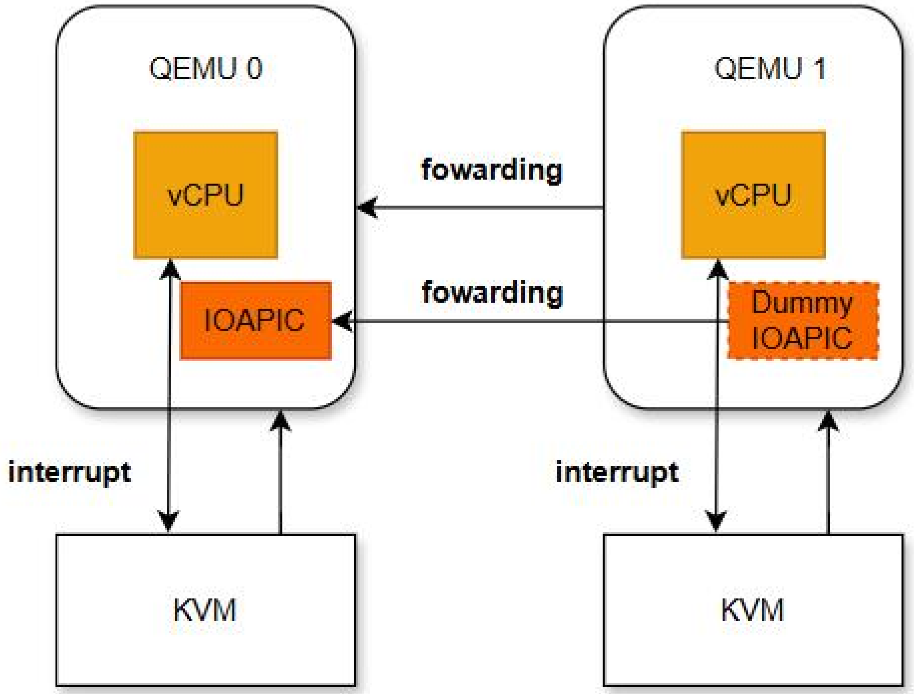
GiantVM: Overview of Architecture

- **Distributed vCPU**
  - Local vCPU and Remote vCPU
  - IPI forwarding
- **Distributed Shared Memory**
  - IVY protocol (for CC)
  - Implemented in EPT
- **Distributed I/O**
  - Same as IPI forwarding
- **Implemented by**

  Qemu 2.8 and Linux 4.8.10 before

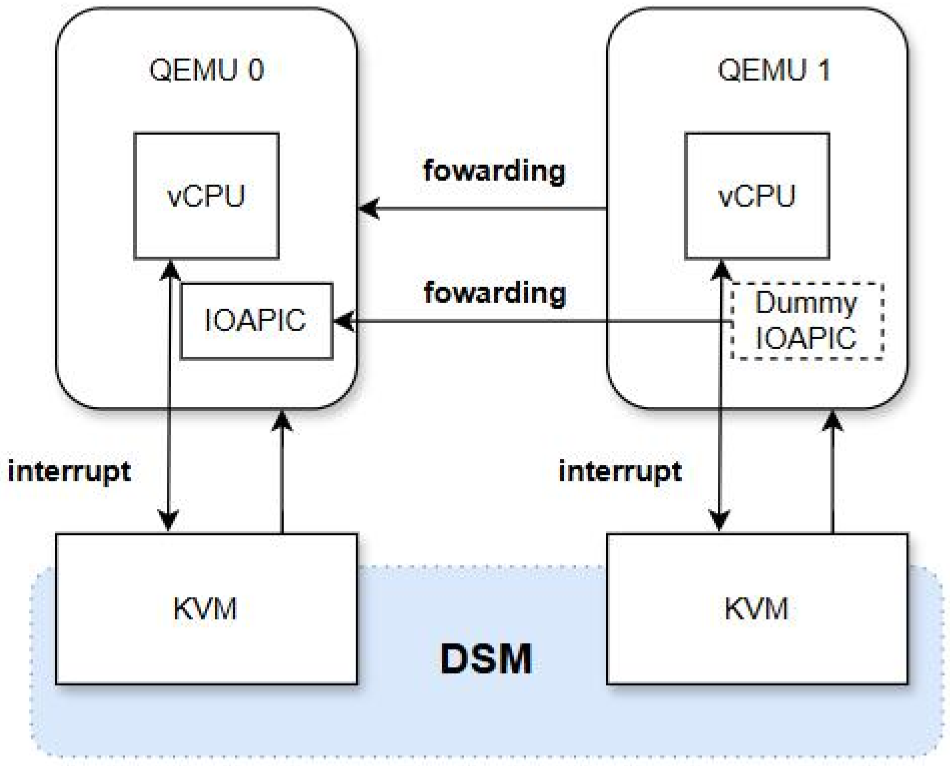


Distributed vCPU and I/O



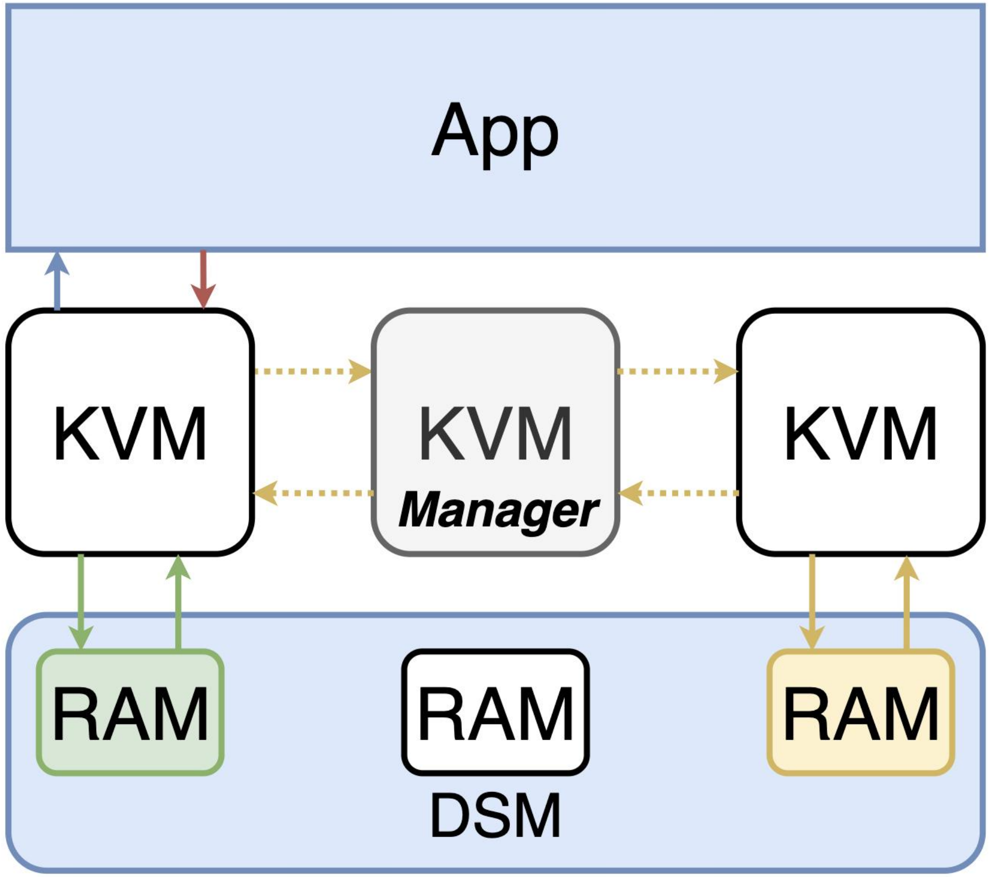
- **Each node run the local vcpu *only***
  - remote vcpu will pend at *qemu_remote_cpu_cond()*
- **Only interactions between different nodes should be considered.**
  - E.g., inter-processor interrupts (IPI), memory-mapped I/O (MMIO), port I/O (PIO), etc
- **GiantVM intercepts them by forwarding instructions to the proper remote node**

Distributed Shared Memory (DSM)



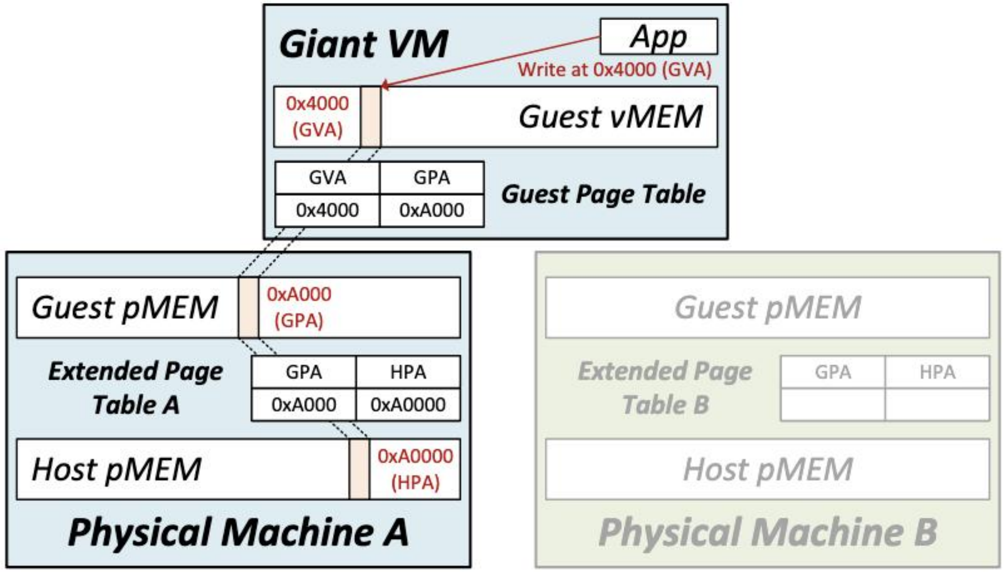
- **IVY protocol for EPT**
  - Each page of memory space has one of three states
    - Modified
    - Shared
    - Invalid

A workflow of DSM fetching

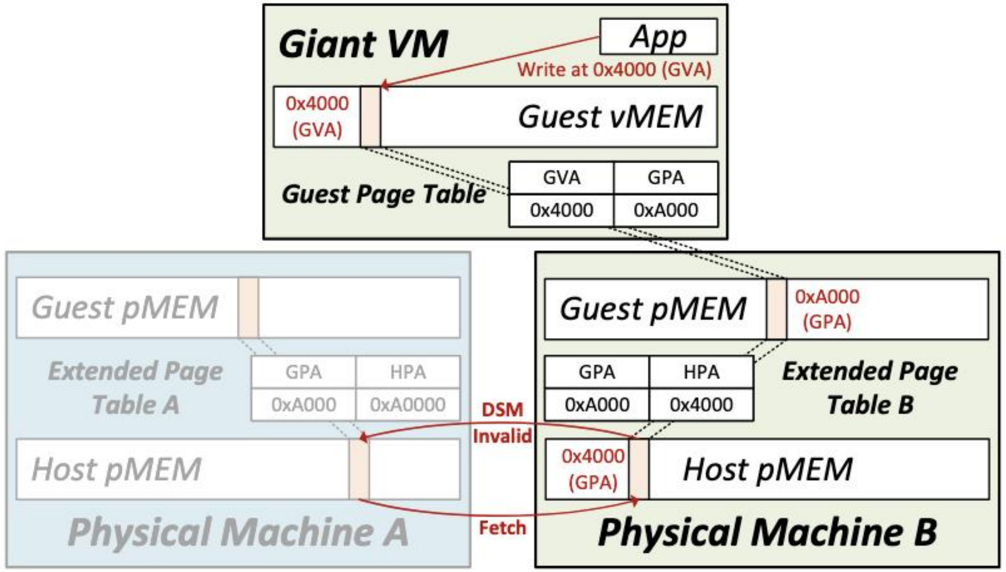


- Guest OS access memory by GPA which then translated by EPT in KVM (red line)
- If the memory is not local, a request will be sent to the manager (who maintains cache coherence), and then forwarded to the memory owner. (yellow line)
- After retrieving the remote memory, it must first be written into the local memory and marked as shared to maintain cache coherence (green line)
- Finally, the node accesses the memory content. (blue line)

DSM Data PINGPONG



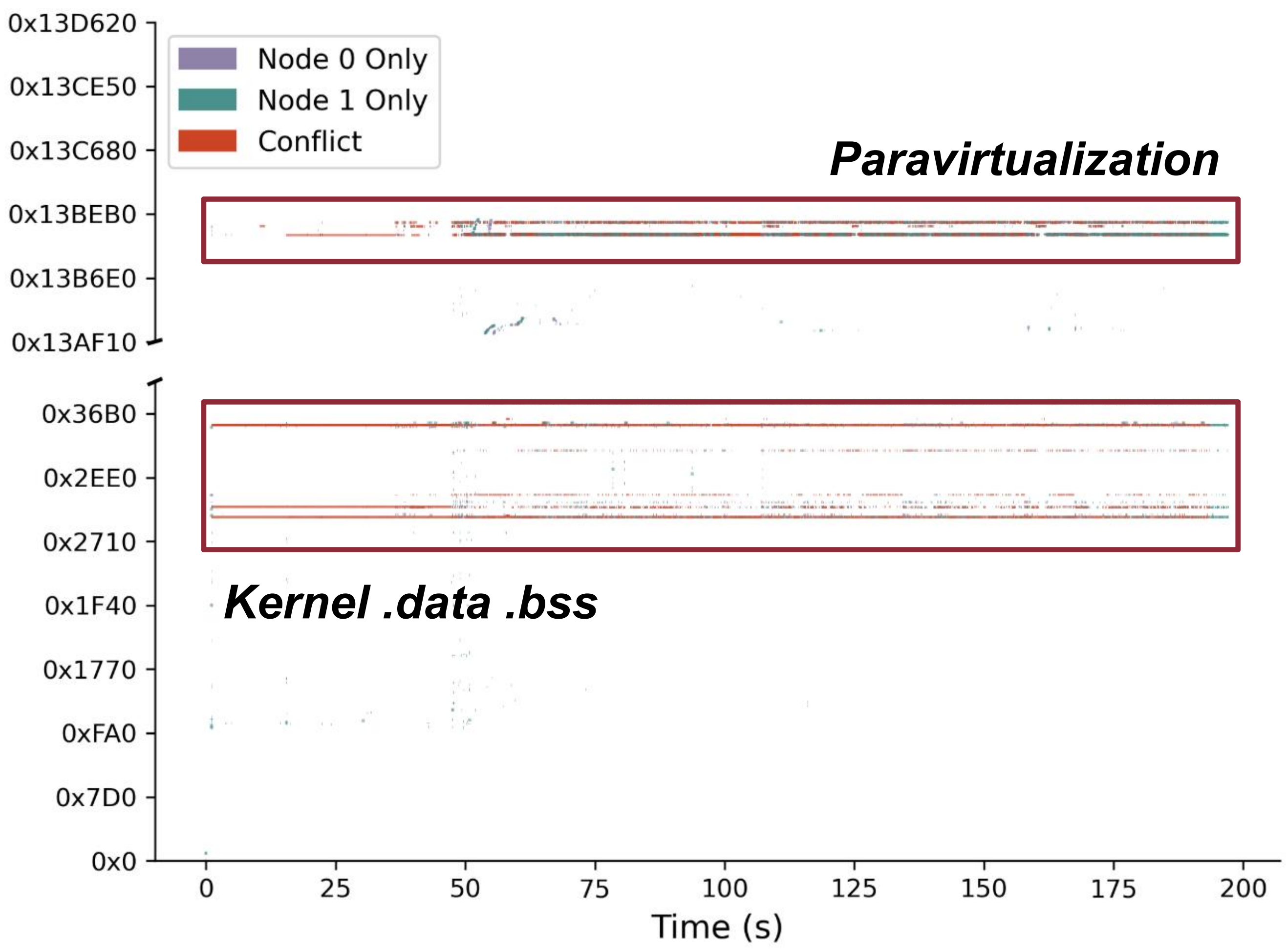
(a) Initial memory write by the application on PM-A.



(b) Cross-node memory access on PM-B by the same application.

- **Software-based DSM -- PING PONG**
  - Applications may access the same GPA on different PM, which would lead to a DSM Page Fault and try to sync data

Case Study -- OS Boot

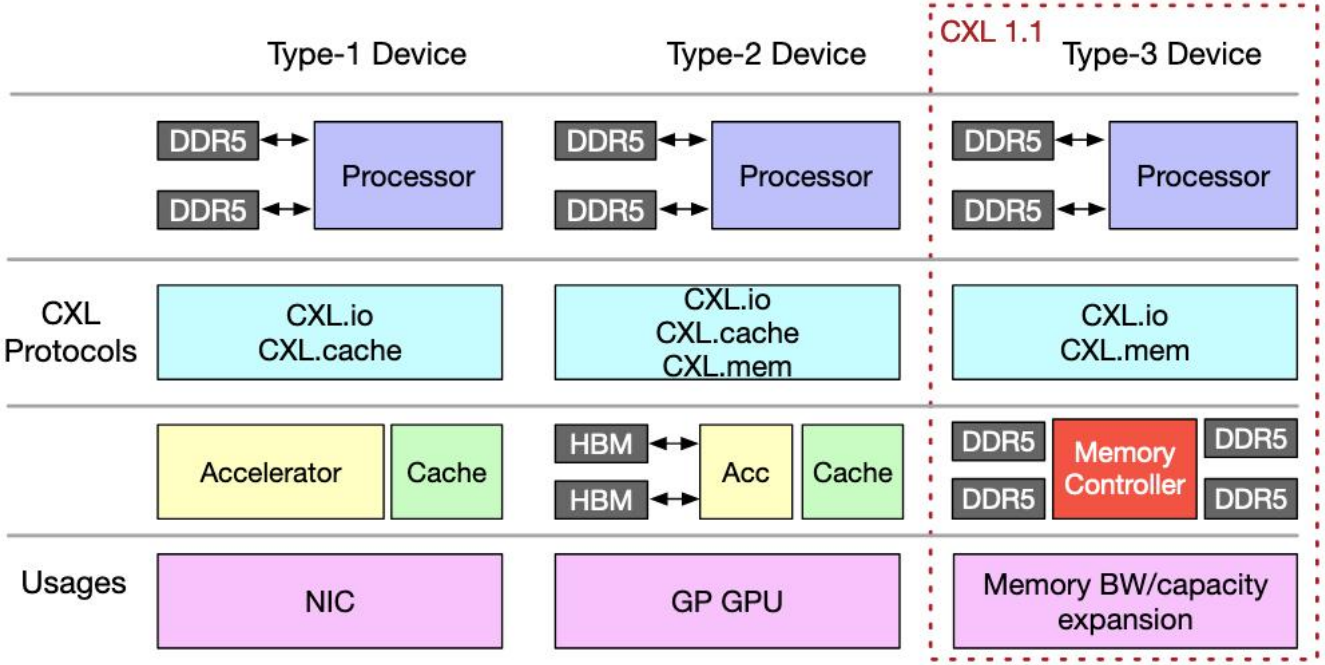


- **OS has shared data**
  - kernel *.data .bss*
  - E.g. jiffies
- **Paravirtualization**
  - E.g. PV EOI

Compute Express Link (CXL)

- **Open standard cache-coherent interconnect for processors, memory expansion, and accelerators.**

- **3 classes of devices:**



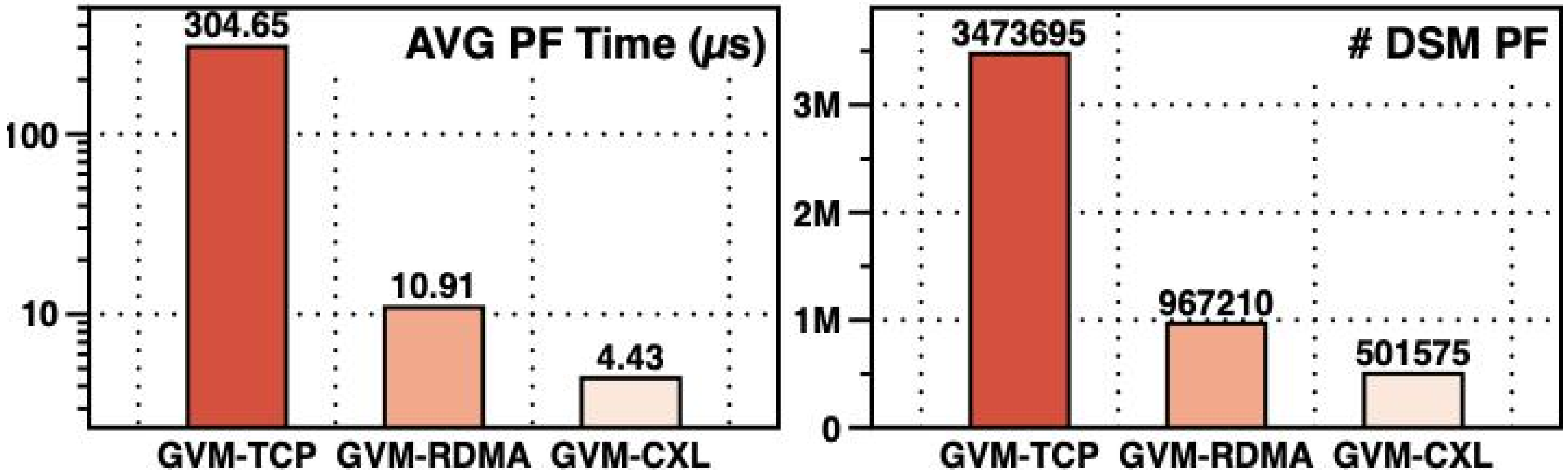
- *Type 1: lack local memeory*
- *Type 2: processor and accelerator can access each memory*
- *Type 3: memory devices*

Compute Express Link (CXL)

- **3 protocols:**
  - CXL.io: Similar to PICe 5.0
    - Init, link up, device discovery and enumeration
  - CXL.cache: Defines interation between host and device
    - enables **accelerators to efficiently access and cache host memory** for optimized performance.
  - CXL.memory: Enables a host, such as a processor, **to access device-attached memory using load/store commands**

CXL for GiantVM

- **Set as daxdev**
- **As a “fast network”**
  - Inspired by HydraRPC, Reduce data sync time from ~10μs (RDMA) to ~5μs
- **Store the “hot” guest page**
  - Boot a 48 cores Guest VM with 10 seconds
  - only store 20 pages



IMPLEMENTATION

Code Porting

- **GiantVM is implemented on Qemu 2 and Linux 4.9, and lacks support for CXL device**
- **We port the code to Qemu 9 and Linux 6.6**

  Sponsored and in collab. by China Telecom Cloud Computing. Thank you!
  - QEMU: ditributed vCPU and I/O
    - **~2800 Loc** modified. Majority of changes are in the newly added **interrupt forwarding module**
  - KVM: distributed shared memory
    - New modules are controlled via Kconfig
    - Most changes are newly added files, with fewer than 400 lines modified in existing files

GiantVM in QEMU

Module/File	Functionality
interrupt-router.c/h	Implements IPI/IOAPIC/x2APIC forwarding across QEMU instances
rdma.c, rdma.h	Support cross-node memory registration and page transfer through RDMA
vl.c, boards.h, qemu-options.hx	Distributed QEMU CLI parsing: \`-local-cpu\`, \`-node-list\`
cpu.h, cpus.c	\`CPUState::local\` tag, remote vCPU wait-count tracking
apic.c, ioapic.c, lapic_internal.h	Local APIC emulation refactored to allow routing to remote CPUs

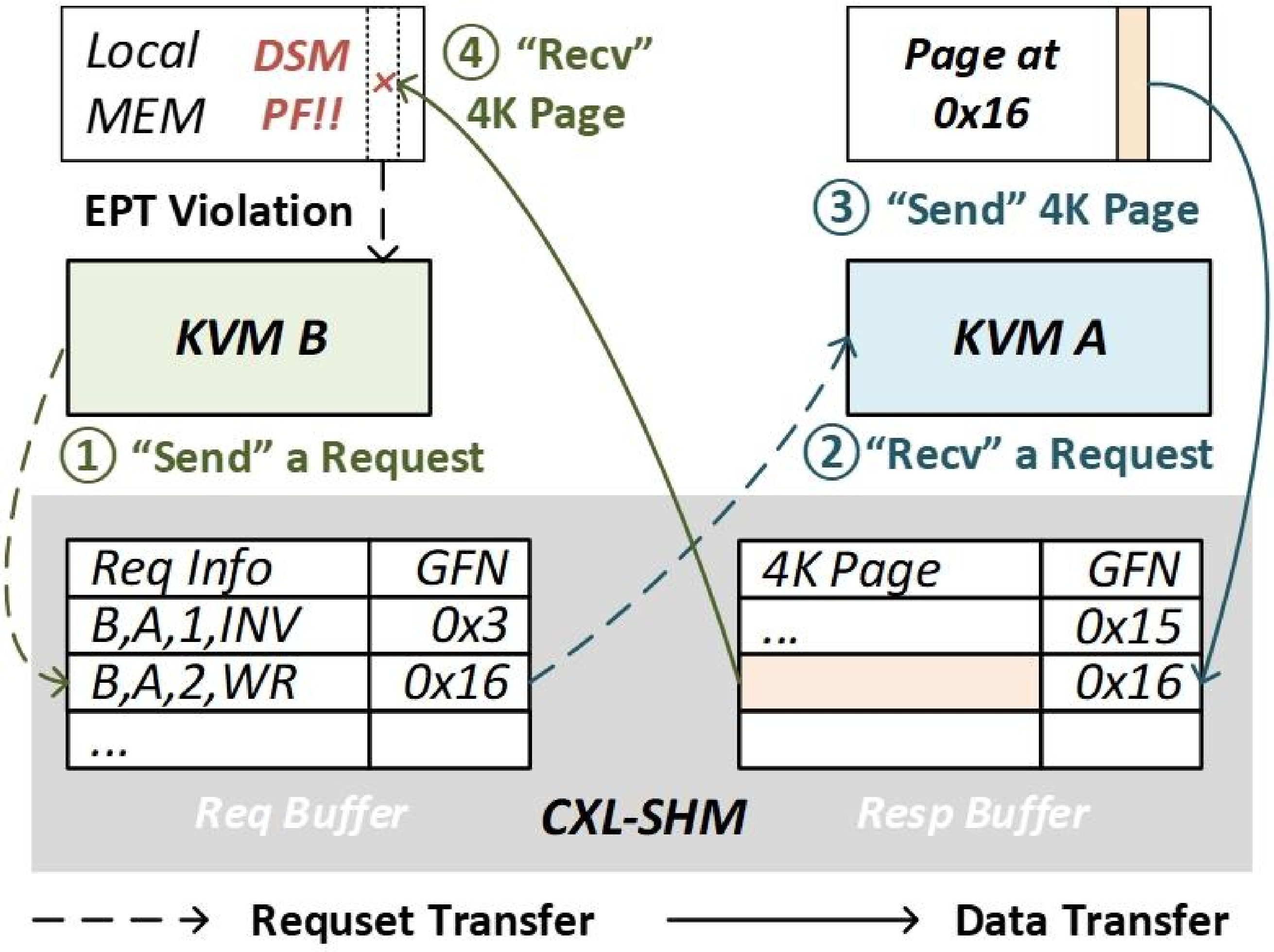
GiantVM in KVM

- **arch/x86/kvm/dsm.c&h,dsm-util.c&h**
  - manage the *kvm_dsm_memslot*
  - Create *kthread* to support Ivy protocol
    - Server thread and client thread
  - ~1500 LOC
- **arch/x86/kvm/mmu/mmu.c,spte.c,lapic.c**
  - spte.c: add dsm-related attributes to the SPTE page (~30 LOC)
  - mmu.c: call the *kvm_dsm_page_fault* to before set spte (~200LOC)
  - lapic.c: call the *kvm_dsm_page_fault* to before pv_eoi (~100LOC)

GiantVM in KVM

- **arch/x86/kvm/ivy.c&h**
  - main function for Ivy protocol
  - ~1000 LOC
- **arch/x86/kvm/ktcp.c&h,krdma.c&h,kcxl.c&h**
  - for data transimission
  - TCP: ~500LOC
  - RDMA: ~1200LOC
  - CXL: ~500LOC

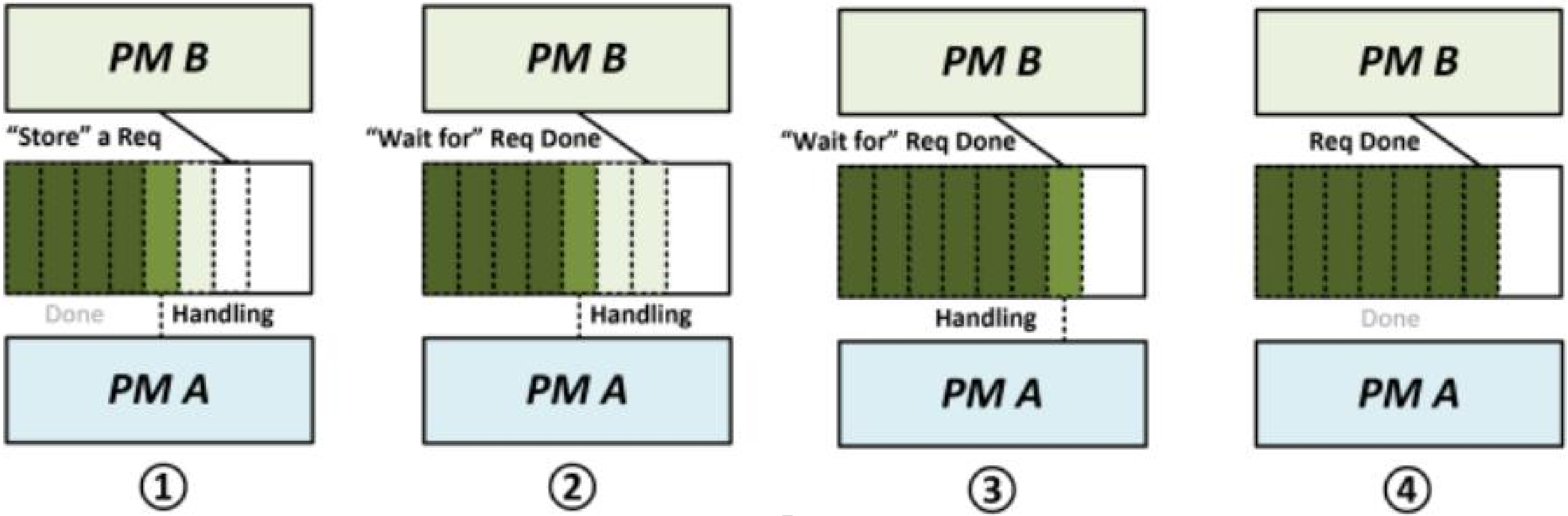
Use CXL as "fast network"



- **Create Req/Resp Buffer for 2 nodes**
  - Send/Recv a request by memcpy
- **Request**
  - ivy transaction
- **Response**
  - 4K page

Handle a Request

- Polling for Req Done
- PCIe MSI can help but need CXL 3.0



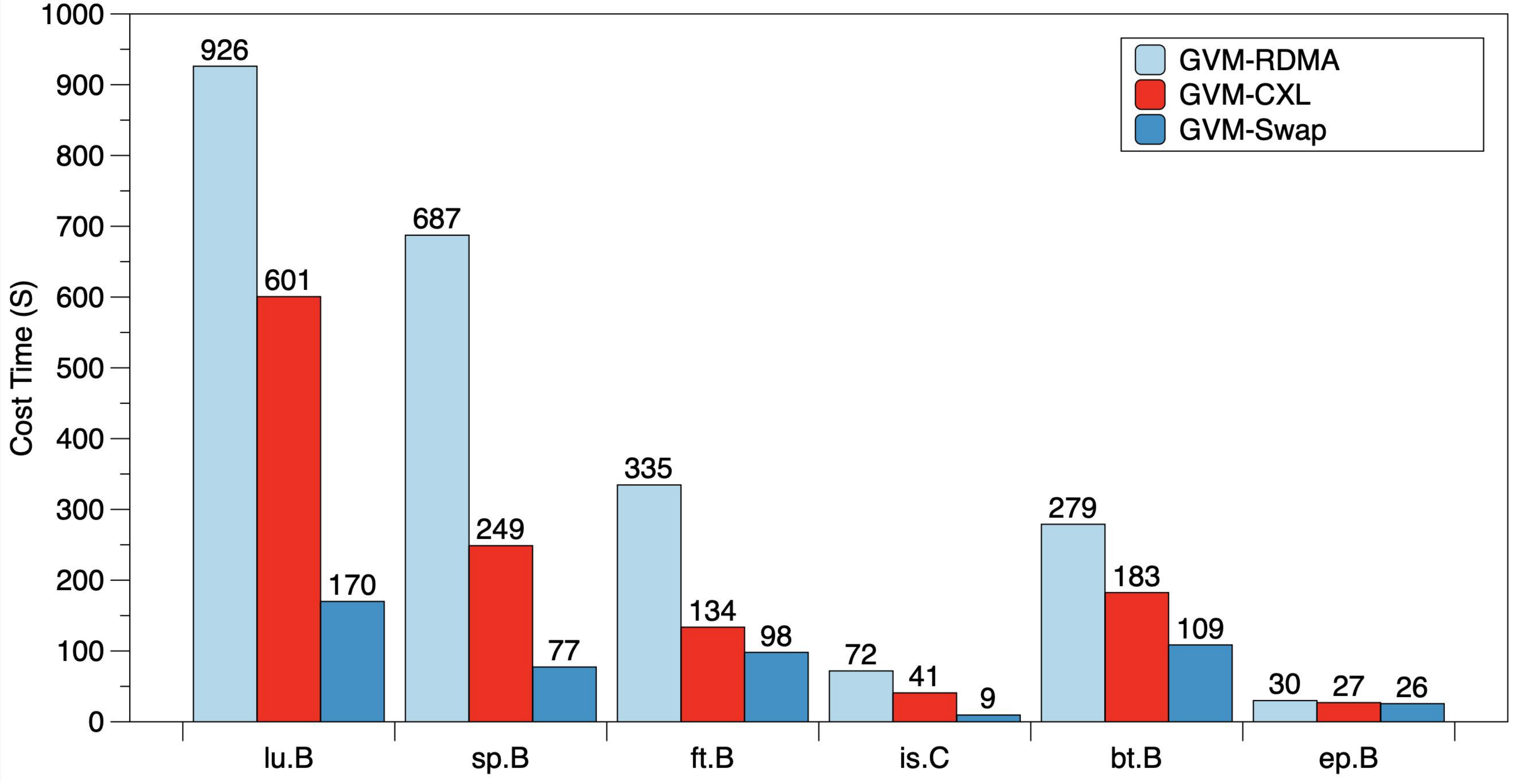
Put “Hot Page” into CXL SHM

- **Change EPT Entry**
  - map the GPA of HOT PAGE (e.g., jiffies) to CXL SHM HPA
- **Memory hierarchy has changed**
  - CXL Shared Memory: ~300ns
  - Softwared-based Shared Memory
    - ~100ns if no DSM PING PONG
    - ~10μs if DSM PING PONG -> detect and swap to CXL-SHM

Put “Hot Page” into CXL SHM

- **Swap to CXL SHM when a write page fault happens**
  - this node would fetch the newest data and set all copyset invalid
  - set itself as owner of this page
  - memcpy to CXL SHM and change the EPT entry
- **When other nodes access this page**
  - trick a page fault because this page is invalid
  - require this page from owner
  - the owner do not send a 4K page but a HPA that the EPT entry should be remapped

Put "Hot Page" into CXL SHM



- **Fast OS boot in <10 seconds, thanks to CXL SHM!**

- **OS boot and Stress-ng demo shown during the talk. Please also check the video content at our website: https://giantvm.github.io/index.html**

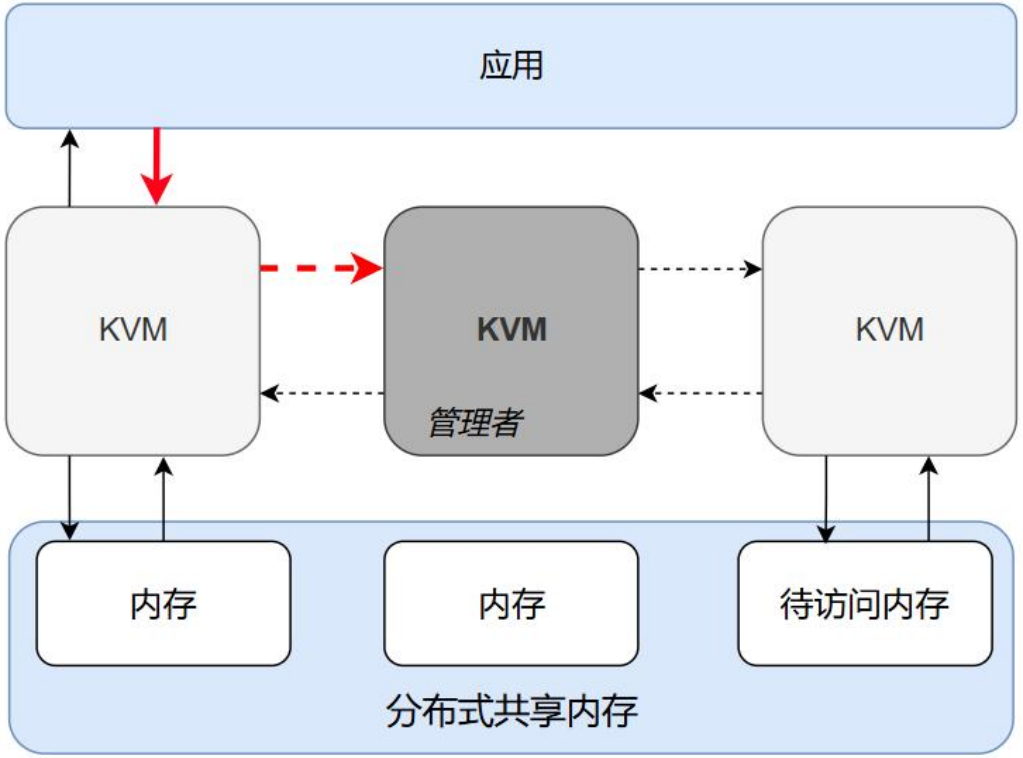
Thank you！

Q&A

The Link to GiantVM:

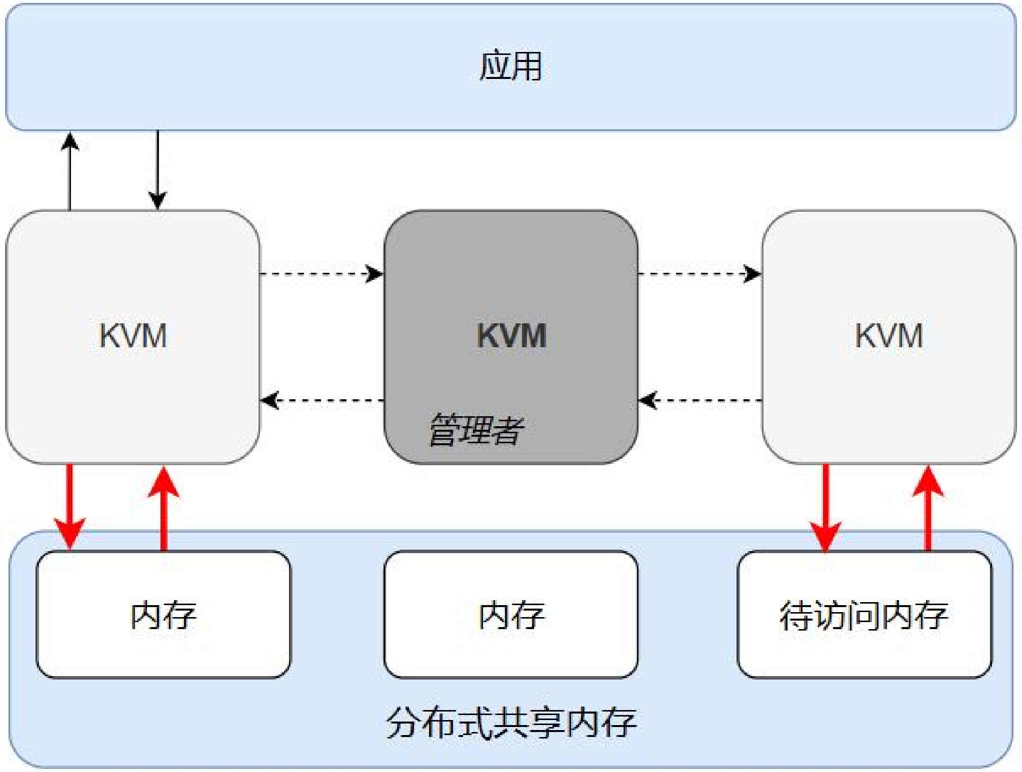
https://giantvm.github.io/index.html

Set EPT



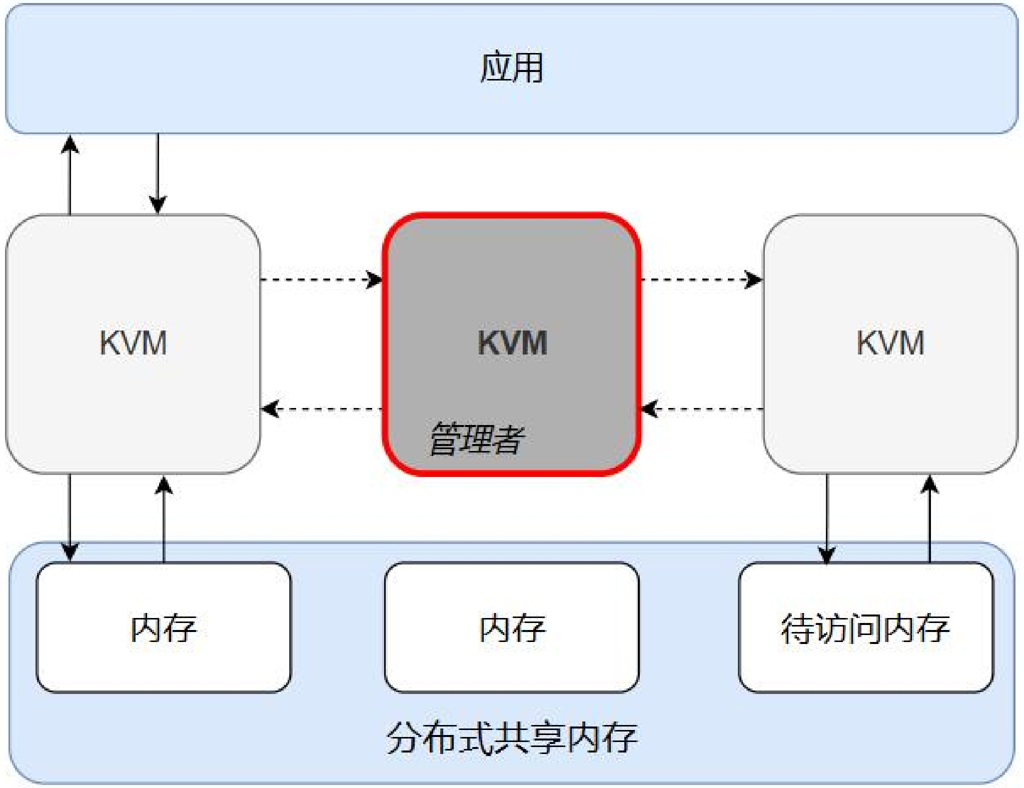
- **页面第一次被访问时，触发缺页异常，并最终建立GFN到PFN的页表项，写入EPT页表**
- **GiantVM在触发缺页异常到写入EPT页表项的过程中，新加入了*kvm_dsm_page_fault*函数，该函数通过内存同步协议，为新建立的页表项设置权限**

Access memory



- KVM通过copy_from_user实现从用户地址空间读写内存
- 对于本地内存，用户地址空间就是QEMU进程的地址空间
- when handle a request from another qemu
  - use_mm to access a non local memory

Trace the memory state



- **各节点间的QEMU进程无法通过虚拟地址进行相互访问，只能通过GFN（Guest Frame Number，客户机页号）**
- **因此，需要维护内存页状态与GFN的映射，以便实现缓存一致性**
- **GiantVM 引入了 *kvm_dsm_memory_slot*，维护了HVA与页表info的映射**
- **同时，内存插槽中维护了GFN到HVA的映射**