CHINA LINUX KERNEL
中国Linux内核开发者大会

vivo

RISC-V Device Shared Work Queue Introduction

Atomic IO Enqueue (AIOE) Extension & AIOE Virtualization (IOMMU GIPC)

GUO REN

Alibaba Damo Academy, XUANTIE Team

CHINA LINUX KERNEL
中国Linux内核开发者大会

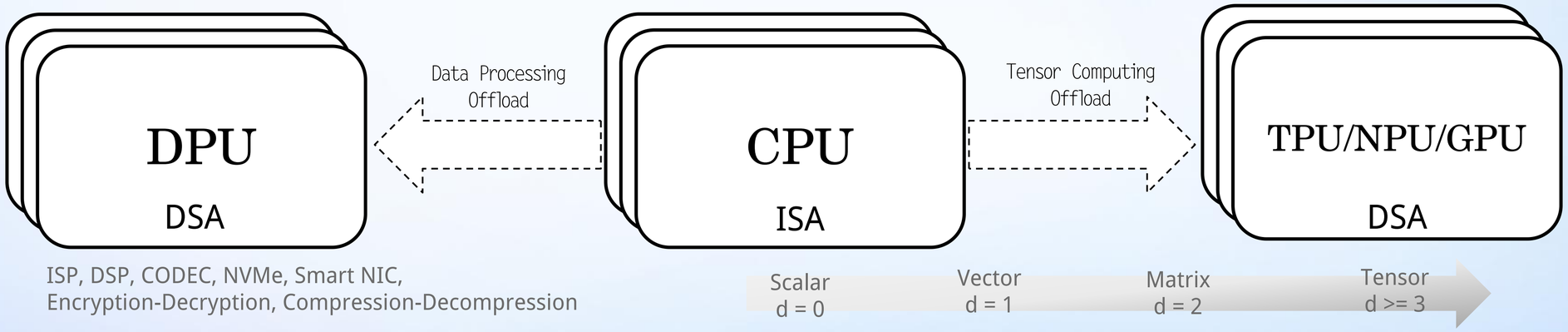
vivo

CONTENTS

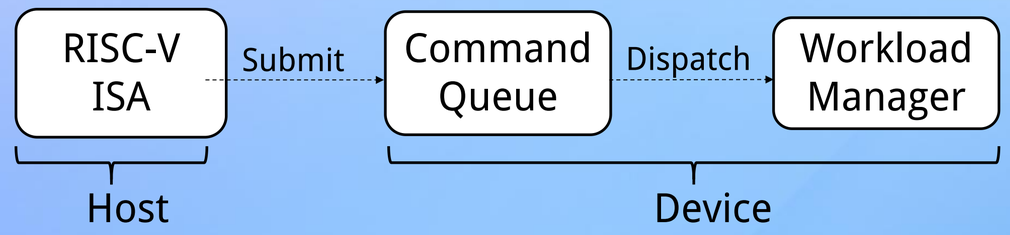
Motivation

Heterogeneous Computing Trends & Problems

Heterogeneous Computing Trends

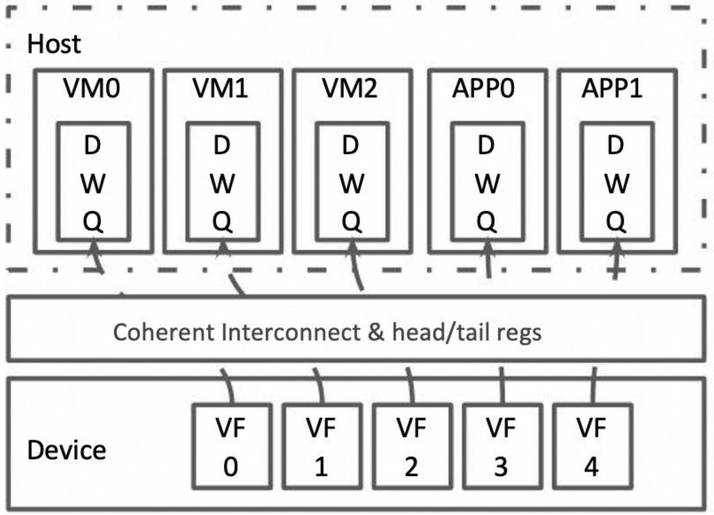


Modern computing systems are heterogeneous.
Thus, the efficiency of heterogenous programming becomes increasingly essential.



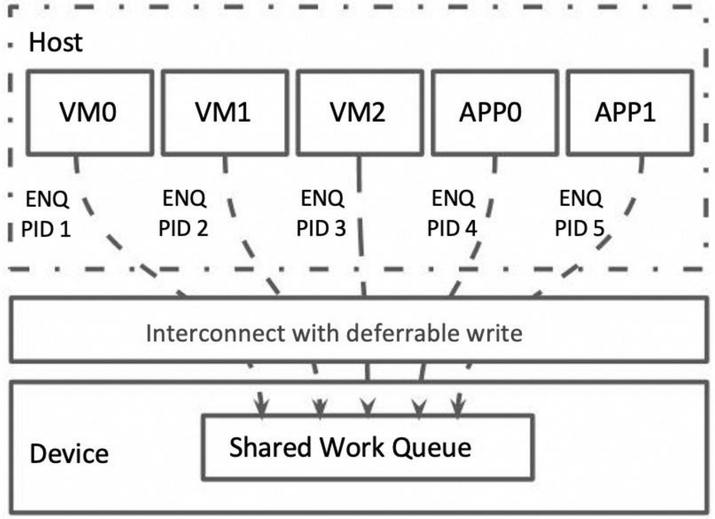
Command Queue Approaches

Dedicated Work Queue



- DWQ resides in Host memory
- Coherency (Cache) interconnect
- Program with head/tail/base IO-regs
- DWQ only serves a single address space

Device Shared Work Queue



- SWQ resides in Device
- Submission with deferrable write transfers (TLP/CHI/AXI)
- Program with ENQ Instruction: ENQCMD, ST64BV0
- SWQ serves with all address spaces by PID
- Return status help control flow

Existing Extensions is Insufficient

- Current ISA lacks Atomic IO Enqueue instructions and its supervisor CSR to append PID (AIOE Draft)
- IOMMU lacks using PID to distinguish VMs (GIPC Draft)
- Interconnects support deferrable write transactions
  - CHI WriteNoSnpDef
  - AXI Deferrable-Write
  - PCI-e DMWr (Deferrable Memory Write) TLP
- Delay problems from PCI-e DMWr status return
  - ROB exhausted entries
  - In-order CPU sensitive to this delay latency

Solution & Proposals

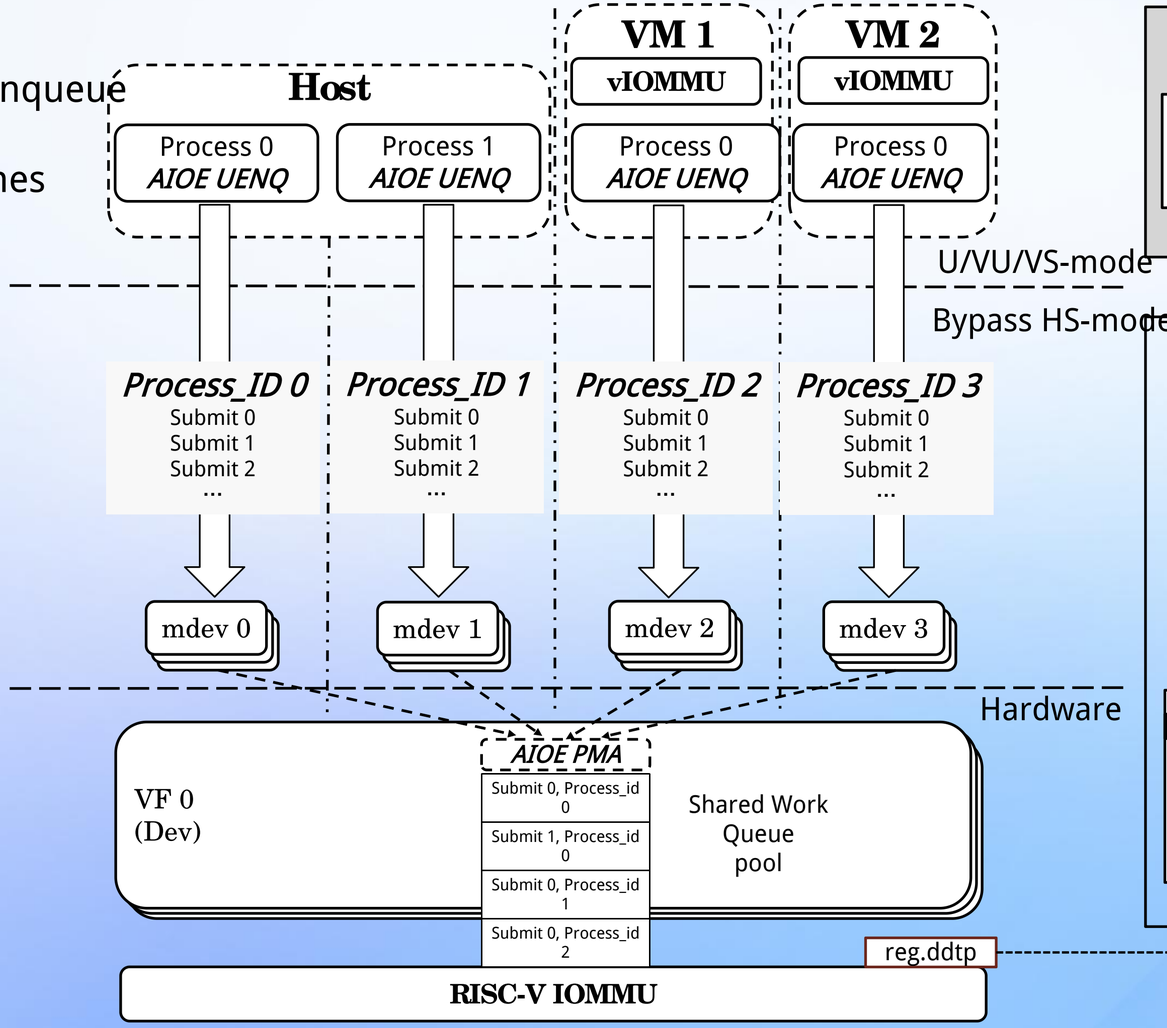
1. Atomic IO Enqueue (AIOE) Extension

2. AIOE under Virtualization:
   G-stage table In Process Context (GIPC) Extension

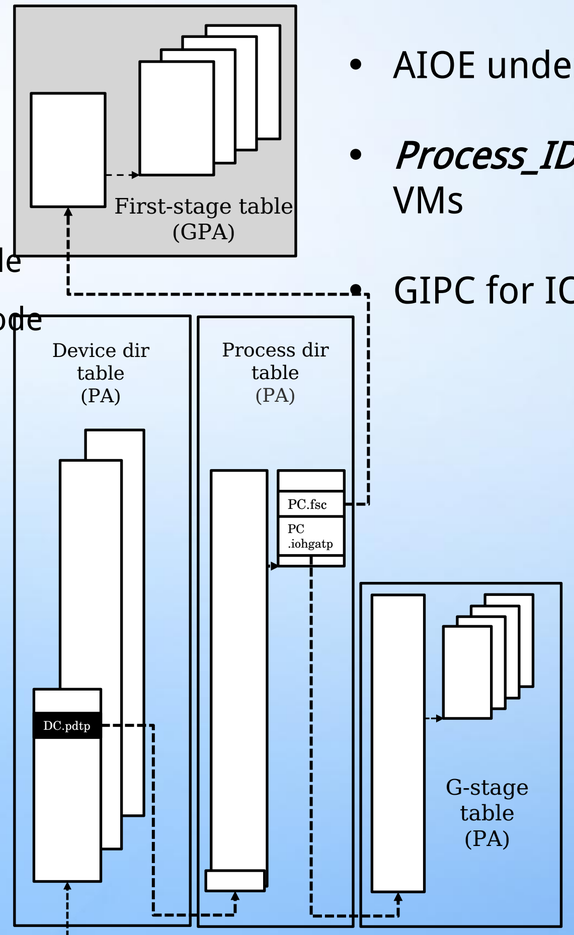
RISC-V Heterogeneous Programming Paradigm

1. Atomic IO Enqueue (AIOE) Extension

- UENQ for Atomic IO Enqueue
- *Process_ID* distinguishes Processes
- AIOE for ISA



2. G-stage table In Process Context (GIPC) Extension



- AIOE under Virtualization
- *Process_ID* distinguishes VMs
- GIPC for IOMMU

Atomic IO Enqueue (AIOE) Extension

UENQ.64B (64-byte Atomic IO Store)

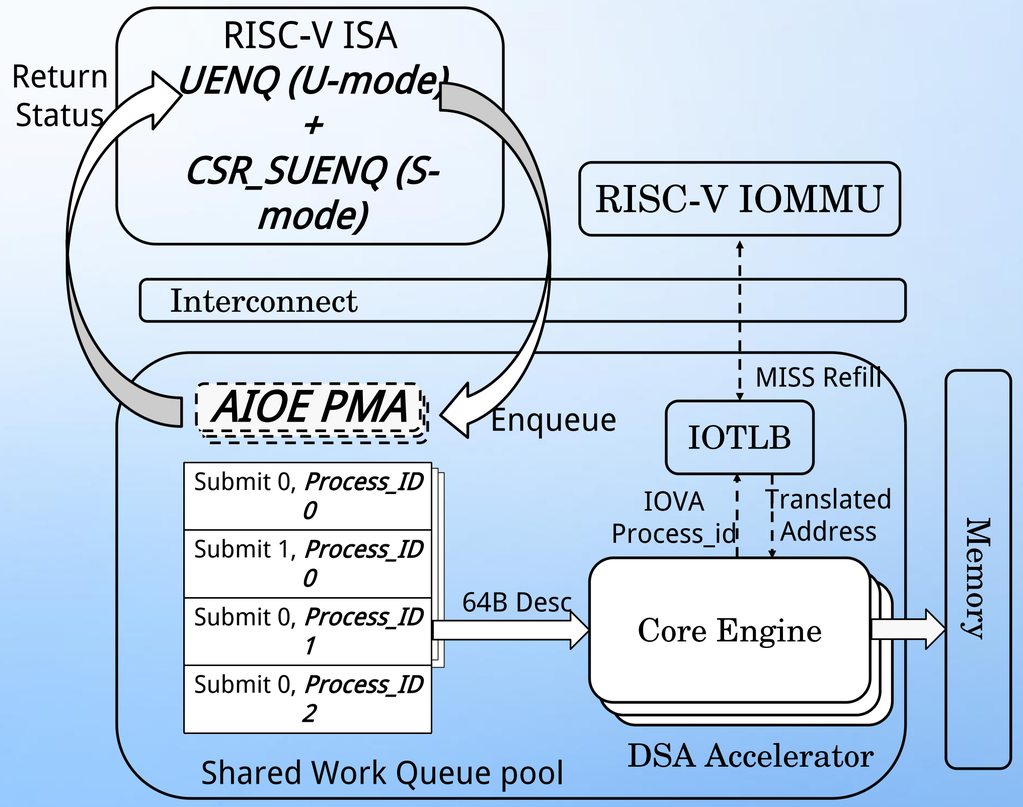
Unprivileged Enqueue (UENQ) instruction of the atomic IO single-write 64-Byte with/without the status result. The 64-Byte store data is formed as data[511:32]:<CSR_SUENQ>[31:0] from 8 consecutive registers.

AIOE PMA (for Deferrable Write Transaction)

Atomic IO Enqueue (AIOE) Physical Memory Attributes (PMA) accepts SENQ and UENQ send-out data.

CSR_SUENQ (for Process_ID)

Privileged CSR register that replaces the lowest bits of the store data of UENQ.64B as Process_ID.



AIOE Summary

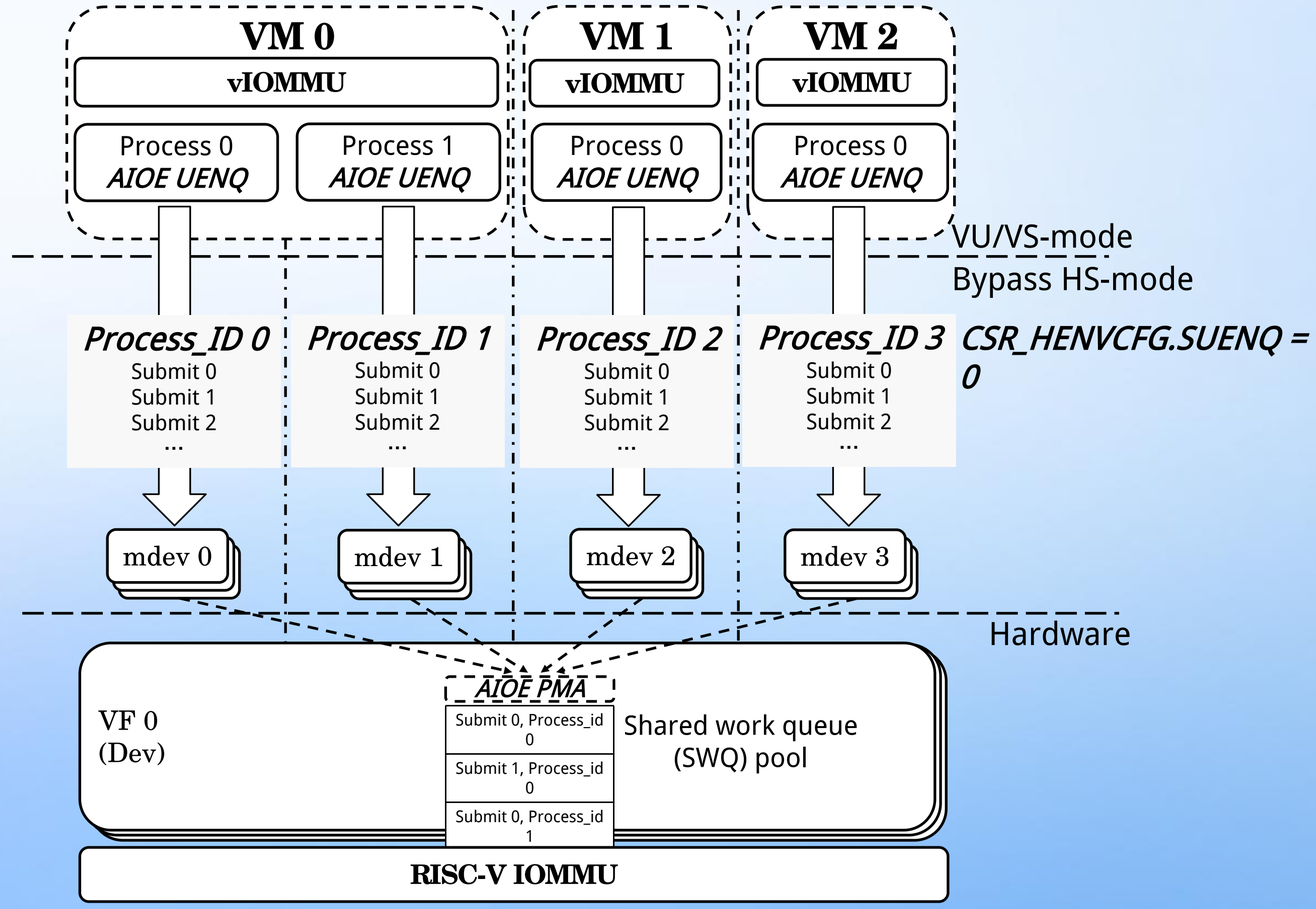
“Atomic IO Enqueue” (AIOE) extension is designed for the RV64 ISA, which contains one PMA definition, two user instructions, two supervisor instructions, one single S-mode CSR, and two envcfg control bits:

AIOE PMA	Atomic IO Enqueue Physical Memory Attribute
CSR_SUENQ	Supervisor Read Write CSR for UENQ instructions
UENQ.64B	User Enqueue Instruction for 64-byte
UENQ.32B	User Enqueue Instruction for 32-byte (Optional)
SENQ.64B	Supervisor Enqueue Instruction for 64-byte
SENQ.32B	Supervisor Enqueue Instruction for 32-byte (Optional)
CSR_MENVCFG.SUENQ	Control bit for SENQ & CSR_SUENQ in S/HS/VS-mode
CSR_HENVCFG.SUENQ	**Control bit for SENQ & CSR_SUENQ in VS-mode (AIOE under Virtualization)**

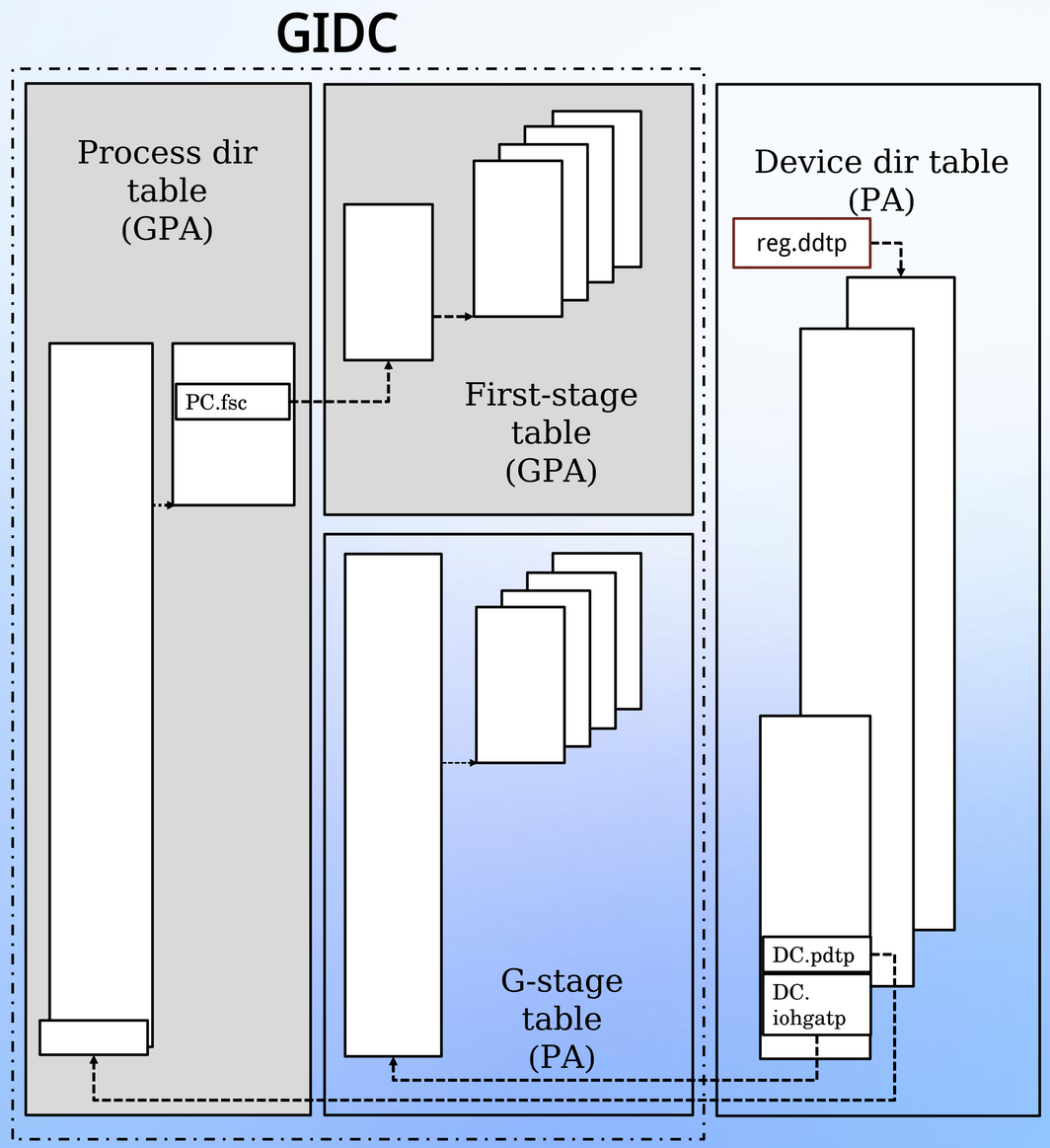
AIOE with Virtualization

- *CSR_HENVCFG.SUENQ = 0*
- *Process_ID distinguishes VMs*
- The vIOMMU maintains First-stage table
- VMM maintains Device dir tables, Process dir tables, G-stage tables

However, the current RISC-V IOMMU distinguishes VM domains by Device_ID:
G-stage table In Device Context (GIDC)



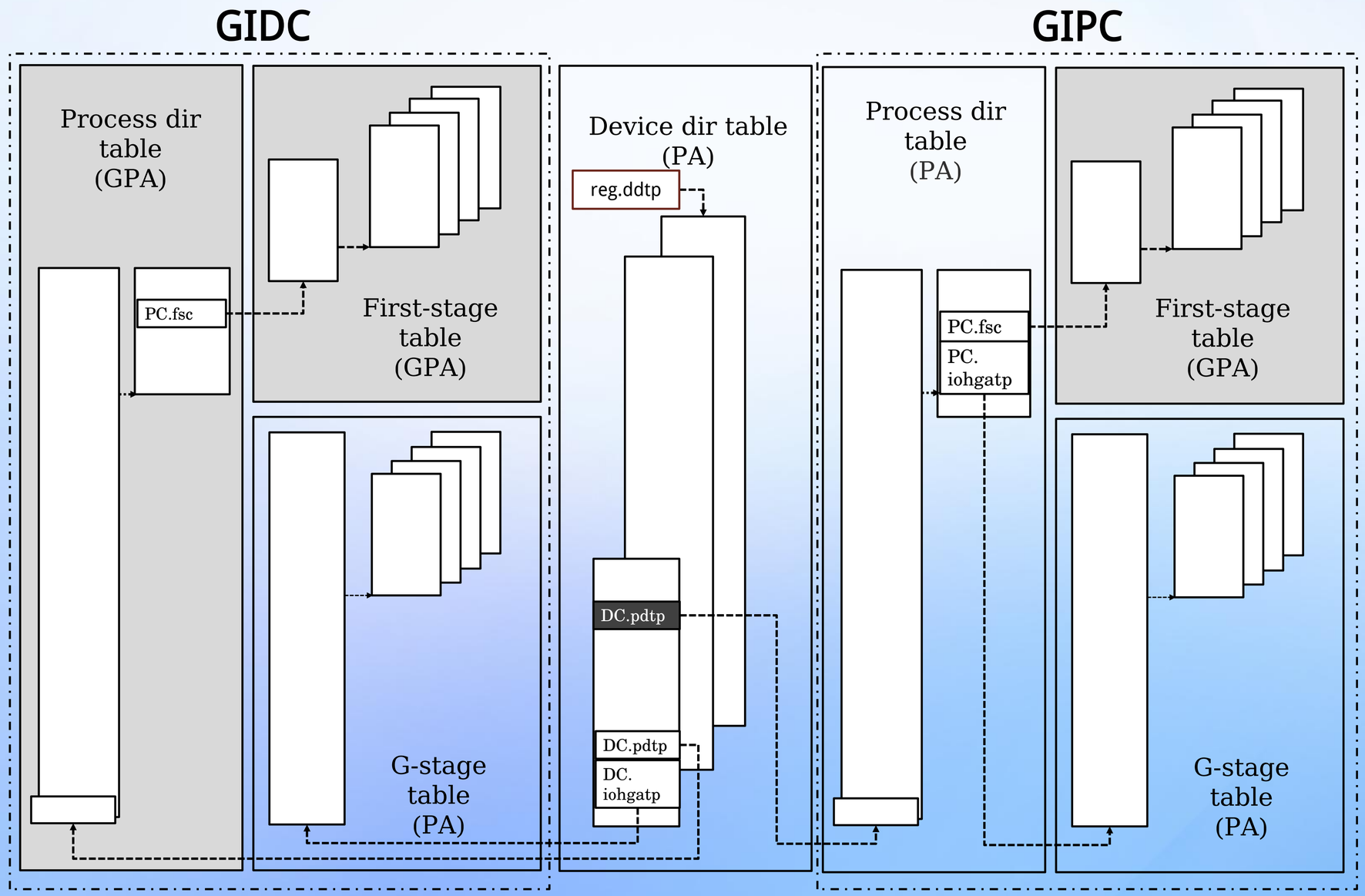
G-stage table In Device Context (GIDC)



VM domain

VMM domain

G-stage table In Process Context (GIPC)



Process_ID distinguishes VMs*:*

- *Process dir table is based on PA and maintained by the VMM*
- *No Nested Page Table Walk*
- *No GPA->PA TLB entries*

VM domain

VMM domain

玄铁分布式 IOMMU 产品介绍

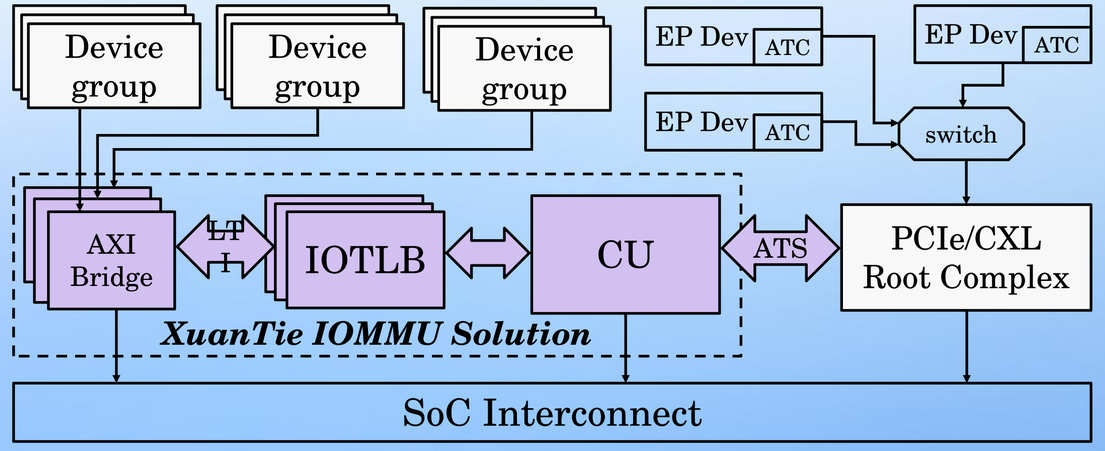
关键硬件特性：

- 分布式高并发 IOTLB 设计，支持 AXI、LTI 灵活集成
- 独立 CU 设计，基于 ATS 适配 PCIe、CXL
- 集成 I/O MPT，支持机密虚拟化
- **面向加速器场景，支持共享队列虚拟化 (GIPC)**
- 支持设备 QoS 管控

- 支持标准 RV IOMMU 规范：
  - DDT (1LVL/2LVL/3LVL)
  - PDT (PD8/PD17/PD20)
  - 1st stage table (Sv39/Sv48/Sv57)
  - 2nd stage table (Sv39x4/Sv48x4/Sv57x4)
  - 1st + 2nd Nested Translation
  - ATS, PRI, T2GPA
  - Atomic PTE (A) and (D) update
  - …

全栈软件生态：

- 支持 Linux IOMMU DMA 驱动框架
- 支持 Linux vfio-pci/mdev 驱动框架
- 支持 Linux vDPA 驱动框架
- 支持 virtio-IOMMU 驱动框架
- 适配 Shared Virtual Address(SVA)应用场景
- 适配 DPDK 和 SPDK
- 支持 CoVE-IO ABI 开源生态和玄铁 XTF、TVM



Final Remarking

Final Remarking

1. **AIOE Extension** eliminates the overhead of synchronization primitives.
   Published at tech-privileged@lists.riscv.org for review v6
   Link: https://lists.riscv.org/g/tech-privileged/message/2320

2. **GIPC Extension** enables AIOE to function across processes of different VMs & domains.
   Link: https://github.com/riscv-non-isa/riscv-iommu/pull/413

Opportunity

Discuss HPC accelerator scenario in datacenter SIG.
Link: https://lists.riscv.org/g/sig-datacenter

Add GIPC ref_model support:

```
Add Test 25 : G-stage table In Process Context (GIPC):

Running IOMMU test suite
Test 01 : All inbound transactions disallowed      : PASS
Test 02 : Bare mode tests                          : PASS
Test 03 : Too wide device_id                       : PASS
Test 04 : Non-leaf DDTE invalid                    : PASS
Test 05 : NL-DDT access viol & data corruption     : PASS
Test 06 : Non-leaf DDTE reserved bits              : PASS
Test 07 : Fault queue overflow and memory fault    : PASS
Test 08 : Device context invalid                   : PASS
Test 09 : Device context misconfigured             : PASS
Test 10 : Unsupported transaction type             : PASS
Test 11 : Dev. ctx. access viol & data corruption  : PASS
Test 12 : Device context invalidation              : PASS
Test 13 : IOFENCE                                  : PASS
Test 14 : G-stage translation sizes                : PASS
Test 15 : G-stage permission faults                : PASS
Test 16 : IOTINVAL.GVMA                            : PASS
Test 17 : S-stage translation sizes                : PASS
Test 18 : S-stage permission faults                : PASS
Test 19 : IOTINVAL.VMA                             : PASS
Test 20 : HPM filtering                            : PASS
Test 21 : Process Directory Table walk             : PASS
Test 22 : ATS page request group response          : PASS
Test 23 : ATS page request                         : PASS
Test 24 : ATS inval request                        : PASS
Test 25 : G-stage table In Process Context (GIPC)  : PASS
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Test 26 : MSI write-through mode                   : PASS
Test 27 : MSI MFIF mode                            : PASS
Test 28 : Illegal commands and CQ mem faults       : PASS
Test 29 : Sv32 mode                                : PASS
Test 30 : Misc. Register Access tests              : PASS

------------------------------------------------------------------------------
                           GCC Code Coverage Report
Directory: .
------------------------------------------------------------------------------
File                                       Lines    Exec  Cover   Missing
------------------------------------------------------------------------------
src/iommu_atc.c                              127     127   100%
src/iommu_ats.c                              163     163   100%
src/iommu_command_queue.c                    215     215   100%
src/iommu_device_context.c                   148     148   100%
src/iommu_faults.c                            41      41   100%
src/iommu_hpm.c                               31      31   100%
src/iommu_interrupt.c                         55      55   100%
src/iommu_msi_trans.c                         75      75   100%
src/iommu_process_context.c                   97      97   100%
src/iommu_reg.c                              399     399   100%
src/iommu_second_stage_trans.c               136     136   100%
src/iommu_translate.c                        231     231   100%
src/iommu_two_stage_trans.c                  174     174   100%
src/iommu_utils.c                              5       5   100%
------------------------------------------------------------------------------
TOTAL                                       1897    1897   100%
------------------------------------------------------------------------------
lines: 100.0% (1897 out of 1897)
```

Objectives

1. Define a standard “RISC-V Device Shared Work Queue Specification” that is composed of many new extensions.
2. Promote the ratification of ISA-related extensions (e.g., AIOE), which will be included in the specification.
3. Promote the ratification of non-ISA-related extensions (e.g., GIPC), which will be included in the specification.
4. Identify gaps in ISA and non-ISA extensions hindering DSWQ implementation, and suggest modification proposals or new extensions to address them. (e.g., DMWr Latency).
5. Deliver Linux-based test infrastructure and test suites, which were developed by "vfio, mdev, iommu" kernel drivers to support userspace drivers. (Qemu -> HW)

backup

Conclusion

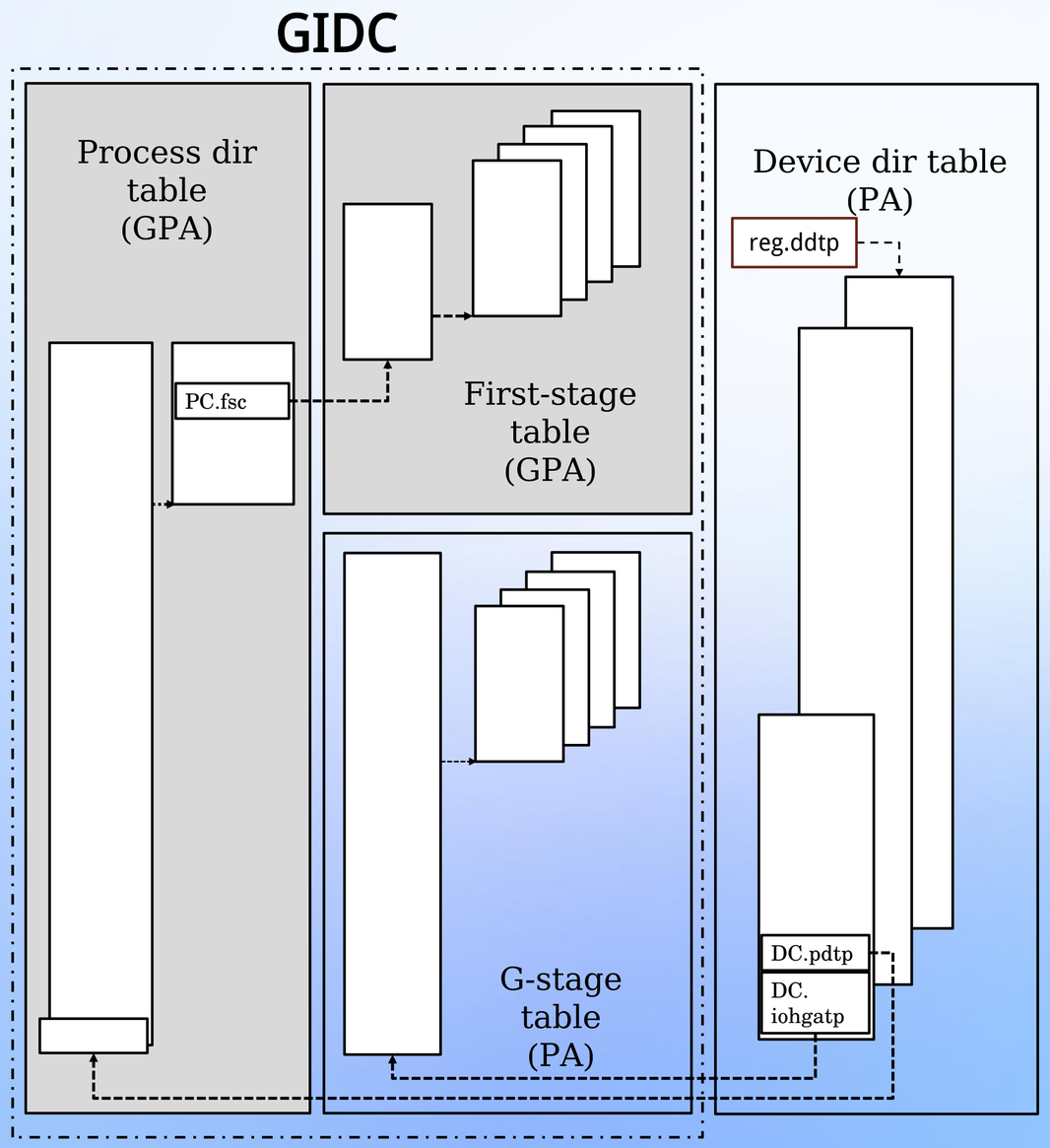
1. **AIOE Extension** eliminates the overhead of synchronization primitives.
2. **GIPC Extension** enables AIOE to function across processes of different VMs & domains.

With the help of AIOE and GIPC, we could explore a new heterogeneous programming paradigm from HPC to embedded scenarios.

Opportunity

Discuss HPC accelerator scenario in datacenter SIG.
Link: https://lists.riscv.org/g/sig-datacenter

RISC-V IOMMU Limitation



Current RISC-V IOMMU implementation:

***G**-stage table **I**n **D**evice **C**ontext (**GIDC**)*

Device_ID distinguishes VMs*:*

- *Process dir table is based on GPA and maintained by the VM*
- *GPA causes Nested Page Table Walk*
- *GPA causes GPA->PA TLB entries*

VM domain

VMM domain

CHINA LINUX KERNEL
中国Linux内核开发者大会

vivo

扫码进入“交流群”

群聊：CLK 2025 大会交流群 2

该二维码7天内(11月8日前)有效，重新进入将更新