Building GPU TEEs using CPU Secure Enclaves with GEVisor

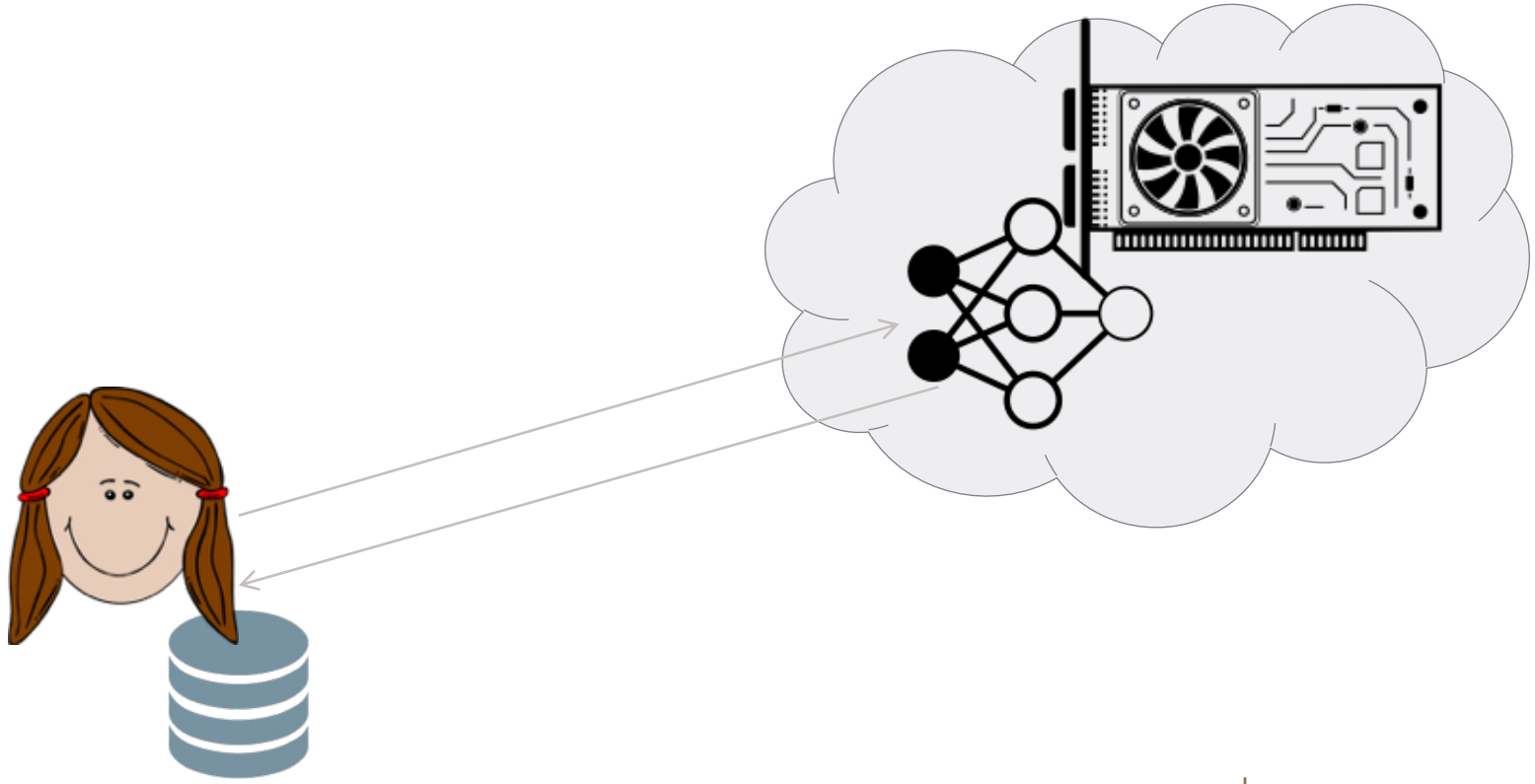
Xiaolong Wu[1], Dave(Jing) Tian[1], Chung Hwan Kim[2]

[1] Purdue University
[2] University of Texas at Dallas

SOCC 2023

Trends in Cloud Computing

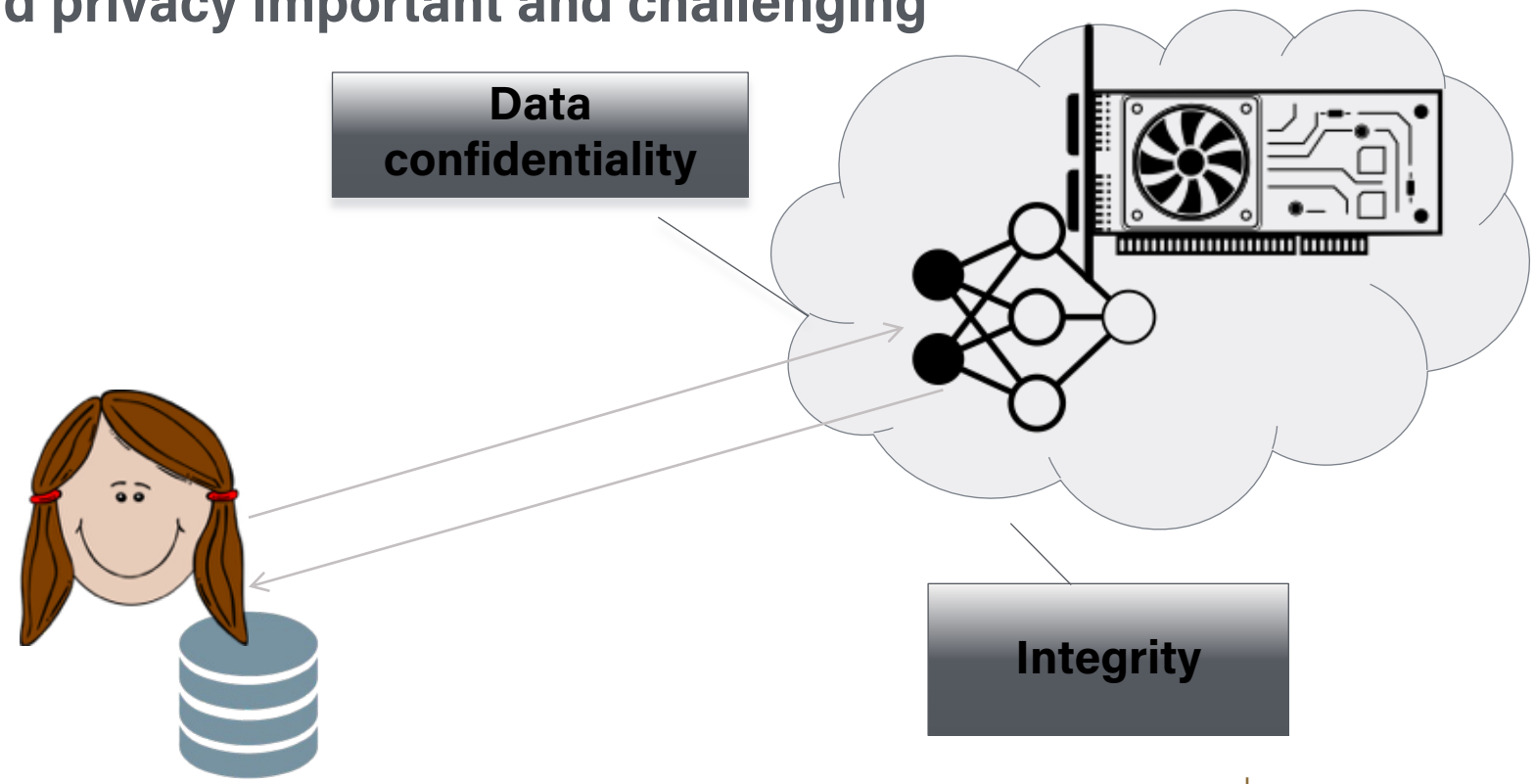
GPU accelerators play important role in cloud



Trends in Cloud Computing

GPU accelerators play important role in cloud

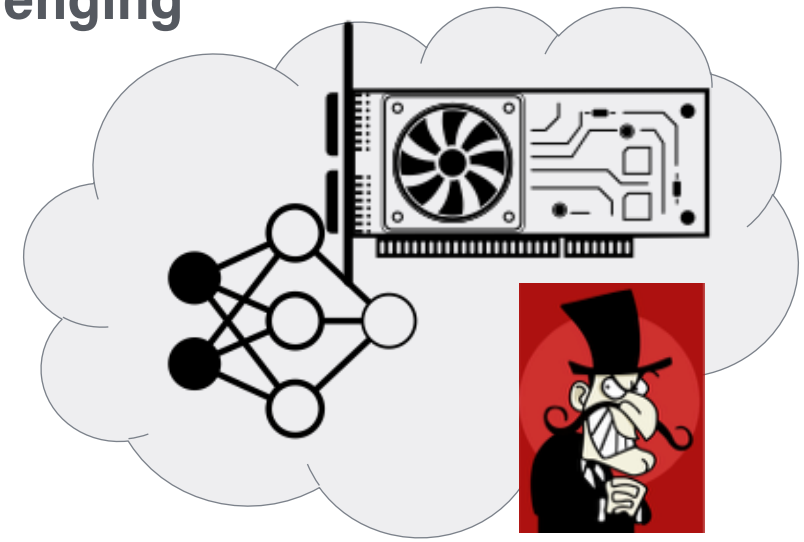
Cloud privacy important and challenging



Trends in Cloud Computing

GPU accelerators play important role in cloud

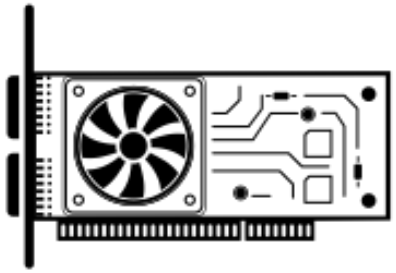
Cloud privacy important and challenging



GPU Trusted Execution Environment (TEE)

Existing hardware GPU TEE solutions prevent current systems from adopting them

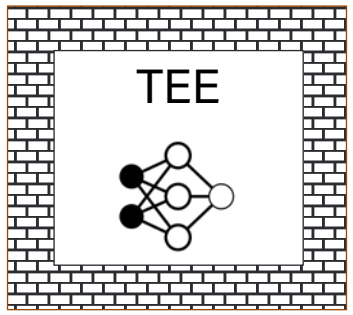
Security flaw found within hardware has to wait for new product to fix it

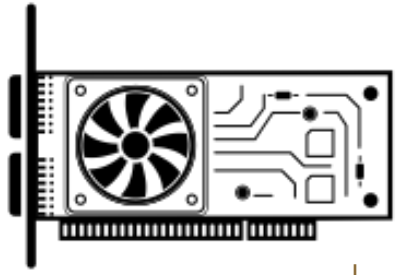


Confidential Cloud Computing

CPU TEEs (e.g., SGX) are prevalent and supported by major cloud providers (e.g., Azure Confidential Computing)

How can we leverage CPU TEE to build GPU TEE practically?

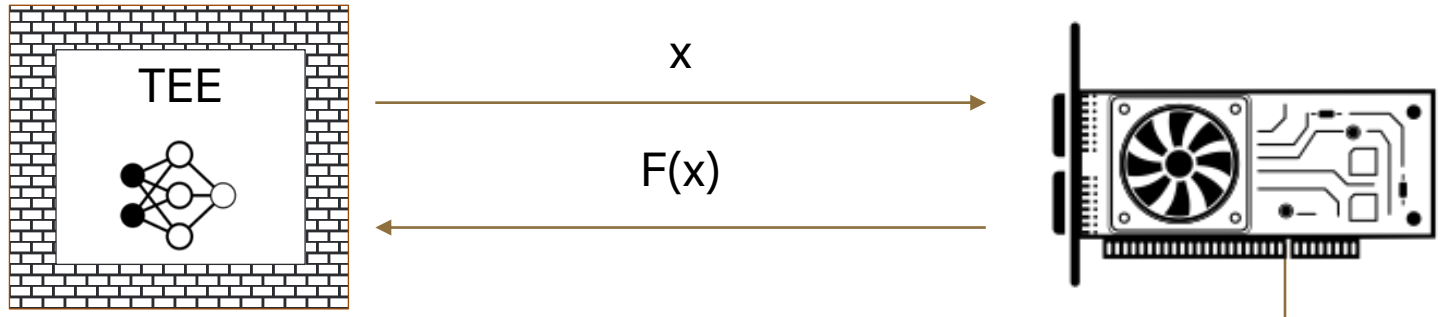




How to Provide TEE to GPU Devices with CPU TEE?

ARM TrustZone supports device I/O protection, but mainly for edge device

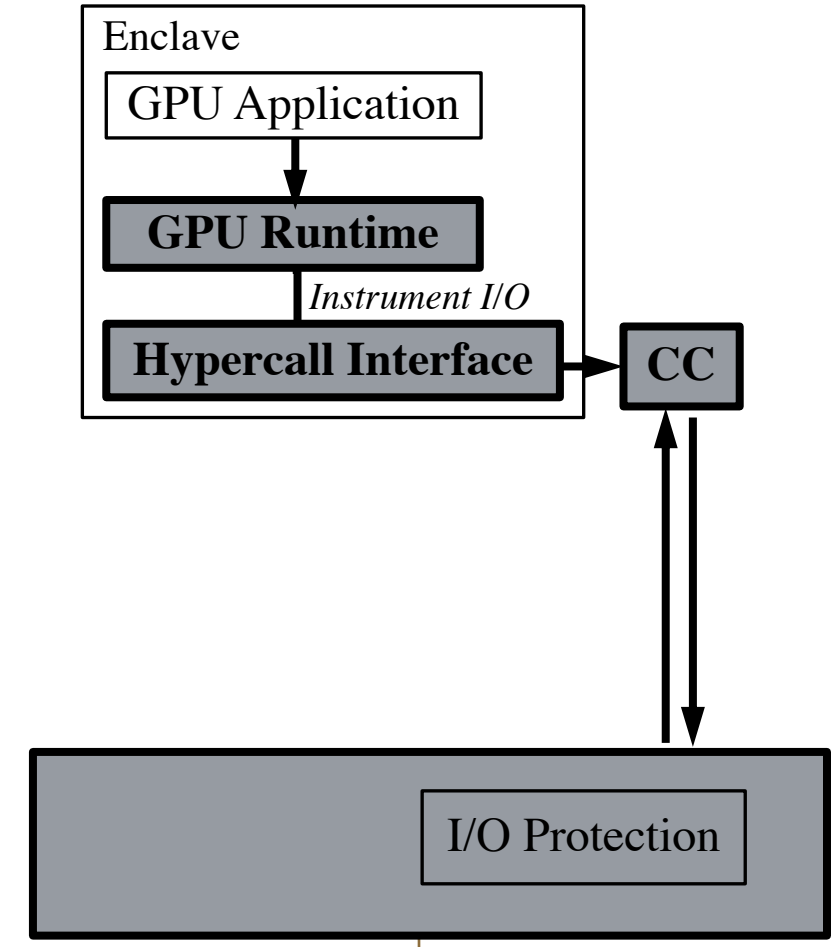
Intel SGX is designed to secure CPU computation, but does not support external device



How do we practically leverage CPU TEEs to Build GPU TEEs

- Idea: Cooperation between enclave and hypervisor
  - Enclave's strong data protection for CPU based attacks
  - I/O protection is handled by Hypervisor

.

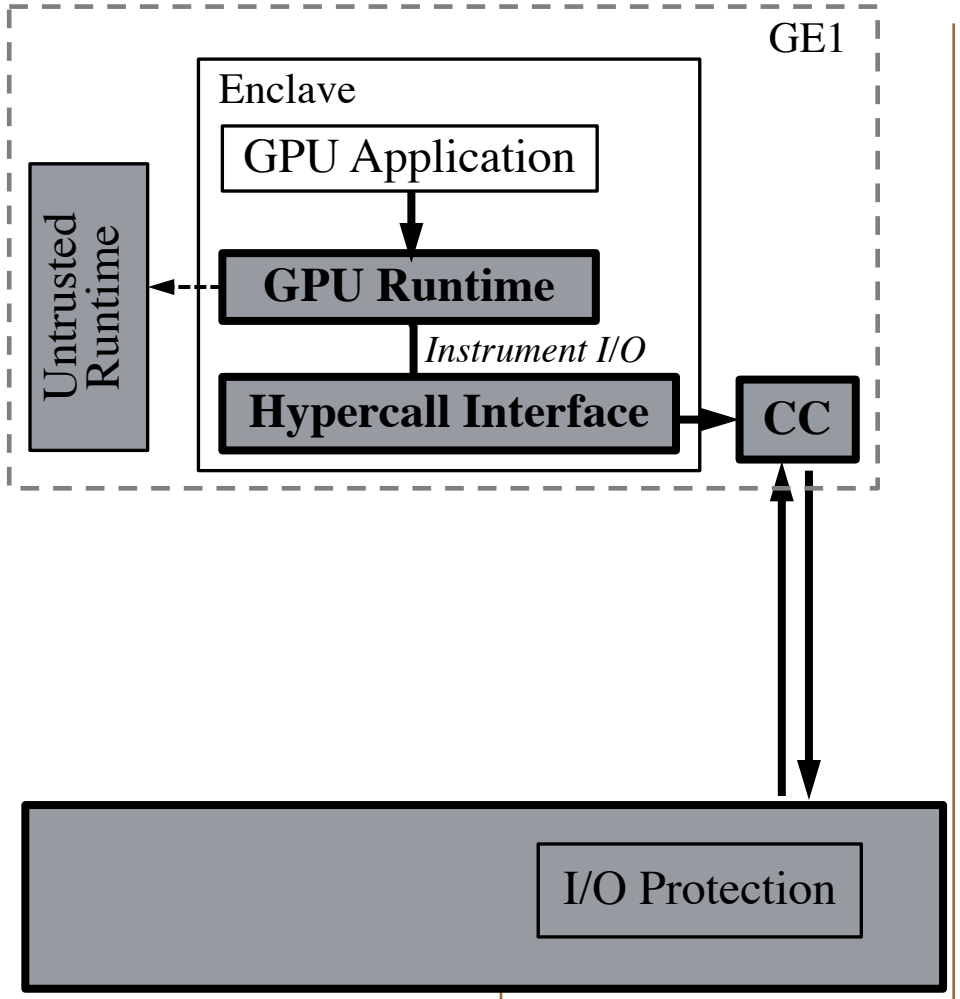


GEVisor

Challenge 1: Trusted I/O

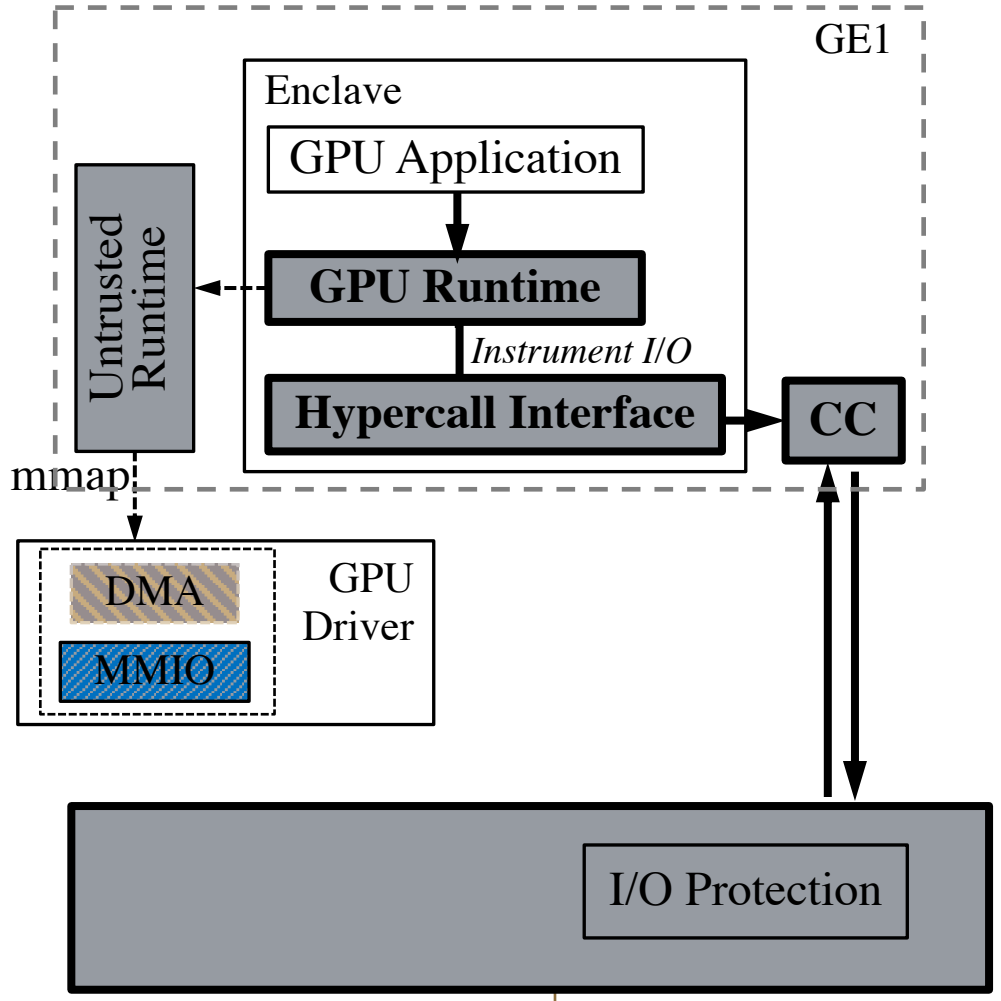
- GEVISOR confines the GPU I/O access and enforces that only the enclave-executing core can access the I/O buffers
- Monitor three events that incur enclave stop running (OCALL, AEX, enclave teardown)

.



Challenge 2: TCB

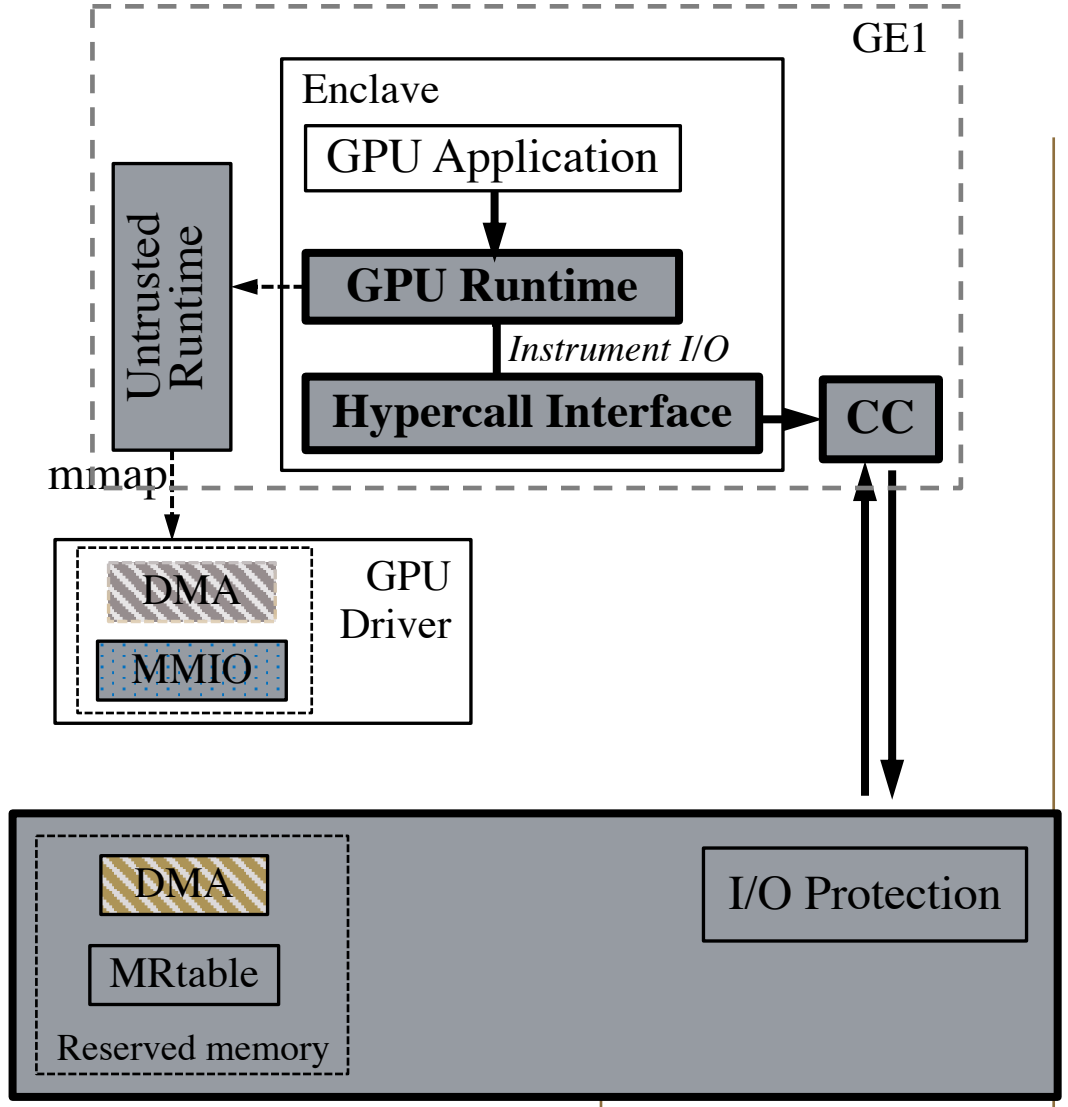
- GPU device driver is large
  (e.g., 1.79 million LoC for AMD GPUs, and 209K LoC for NVIDIA GPU driver (nouveau)).



Challenge 2: TCB

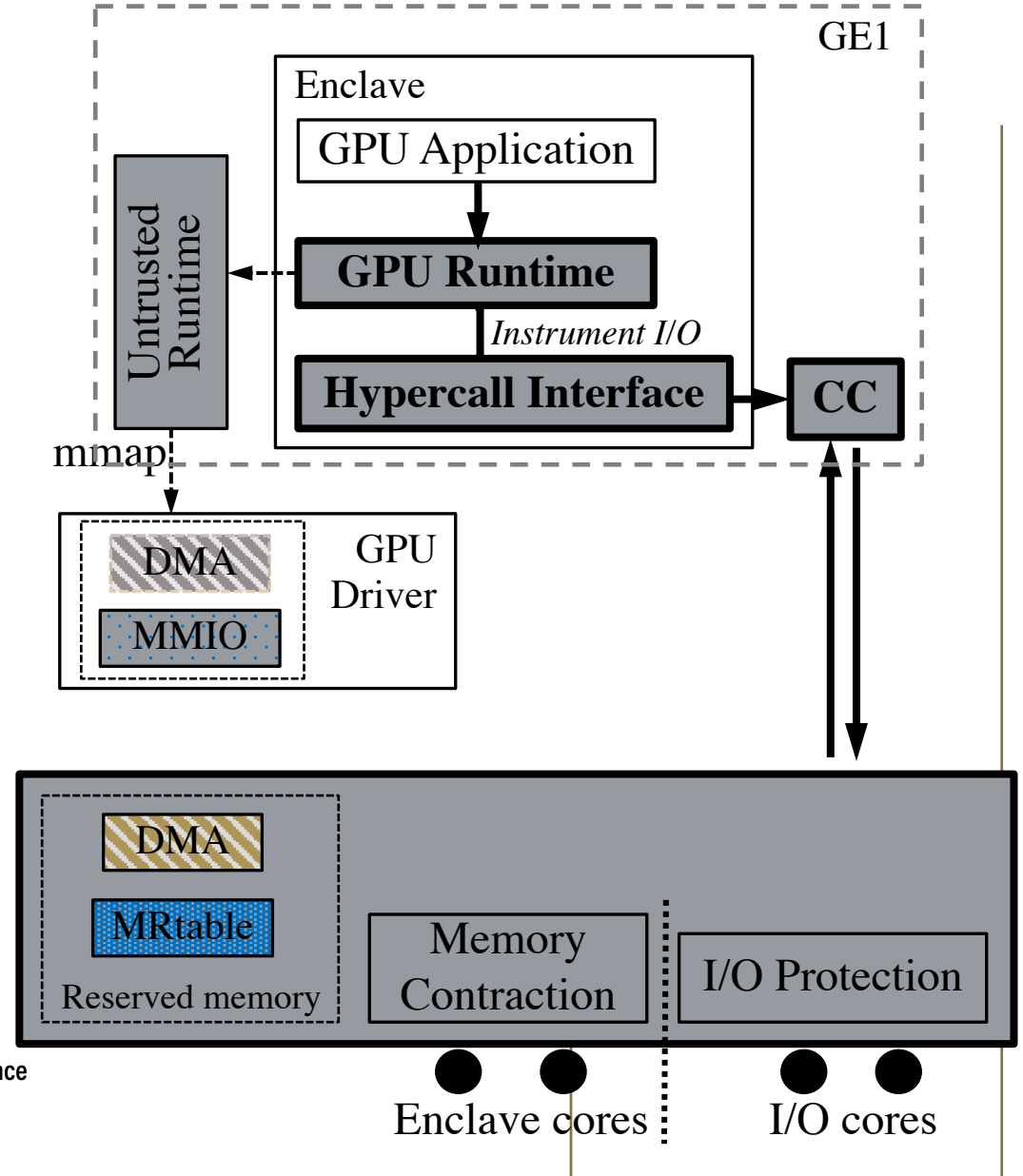
- GPU device driver is large
  (e.g., 1.79 million LoC for AMD GPUs, and 209K LoC for NVIDIA GPU driver (nouveau)).

- GEVisor
  - Hypervisor maintains trusted I/O buffers (MMIO and DMA)



Challenge 3: Performance

- Overhead:
  - Cryptographic primitives are expensive
  - Hypervisor has context switch overhead
- GEVisor
  - Unified MMIO and DMA protection
  - Replace encryption and hashing with EPT
  - Reserved I/O cores for async hypercall



Challenge 4: New attack surface

- Problems:
  - A new GPU TEE solution might still suffer from typical memory safety issues and/or incomplete protections
- Solutions:
  - Formally verify the confidential, integrity, and isolation security protection property of GEVisor with Non-interference policy

GPU I/O access control

- Reserved Mrtable (ID, PA, VA, Size)
- Pass (ID, VA, Size)

Memory Region table (MRtable)

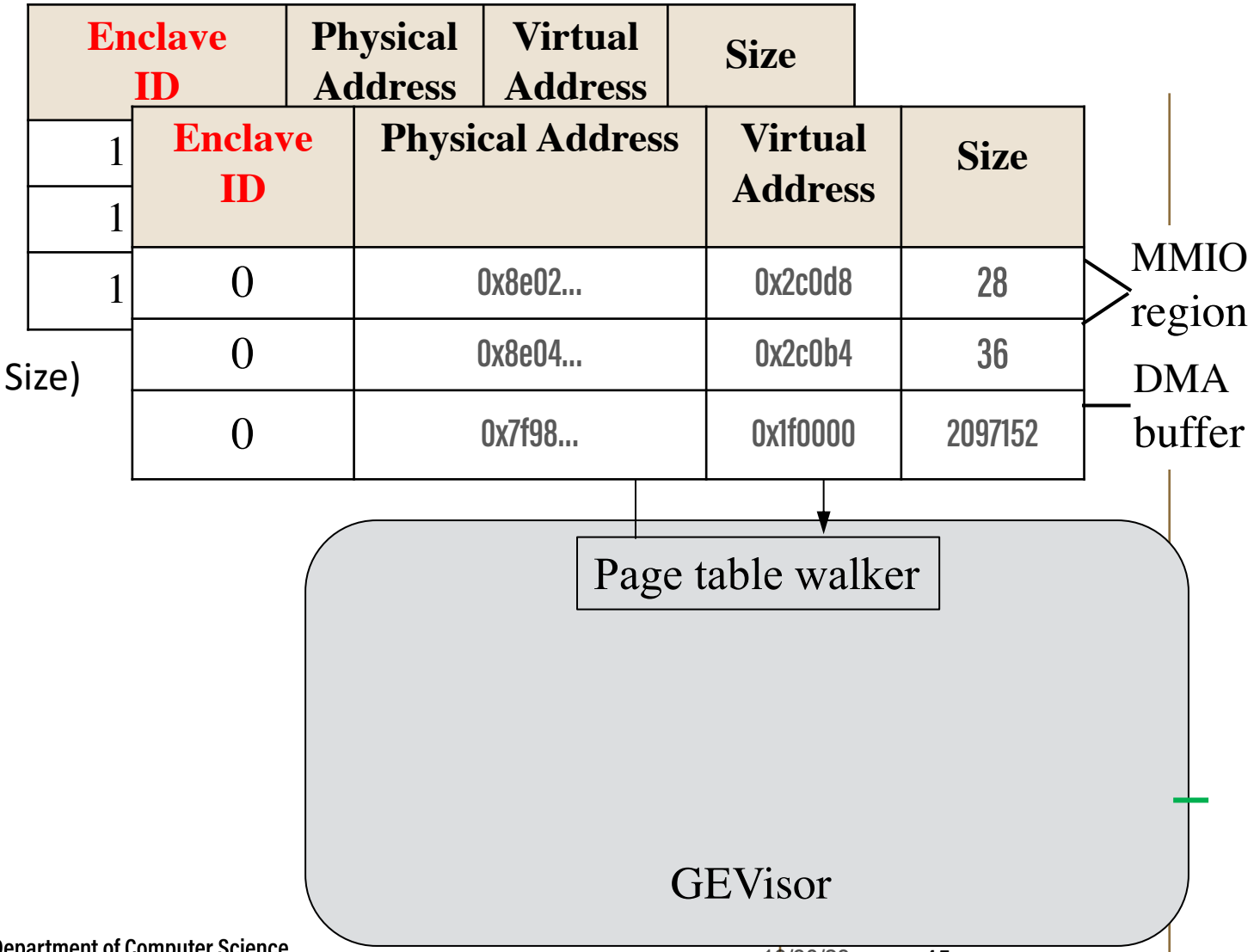
Enclave ID	Physical Address	Virtual Address	Size
0		0x2c0d8	28
0		0x2c0b4	36
0		0x1f0000	2097152

MMIO region

DMA buffer

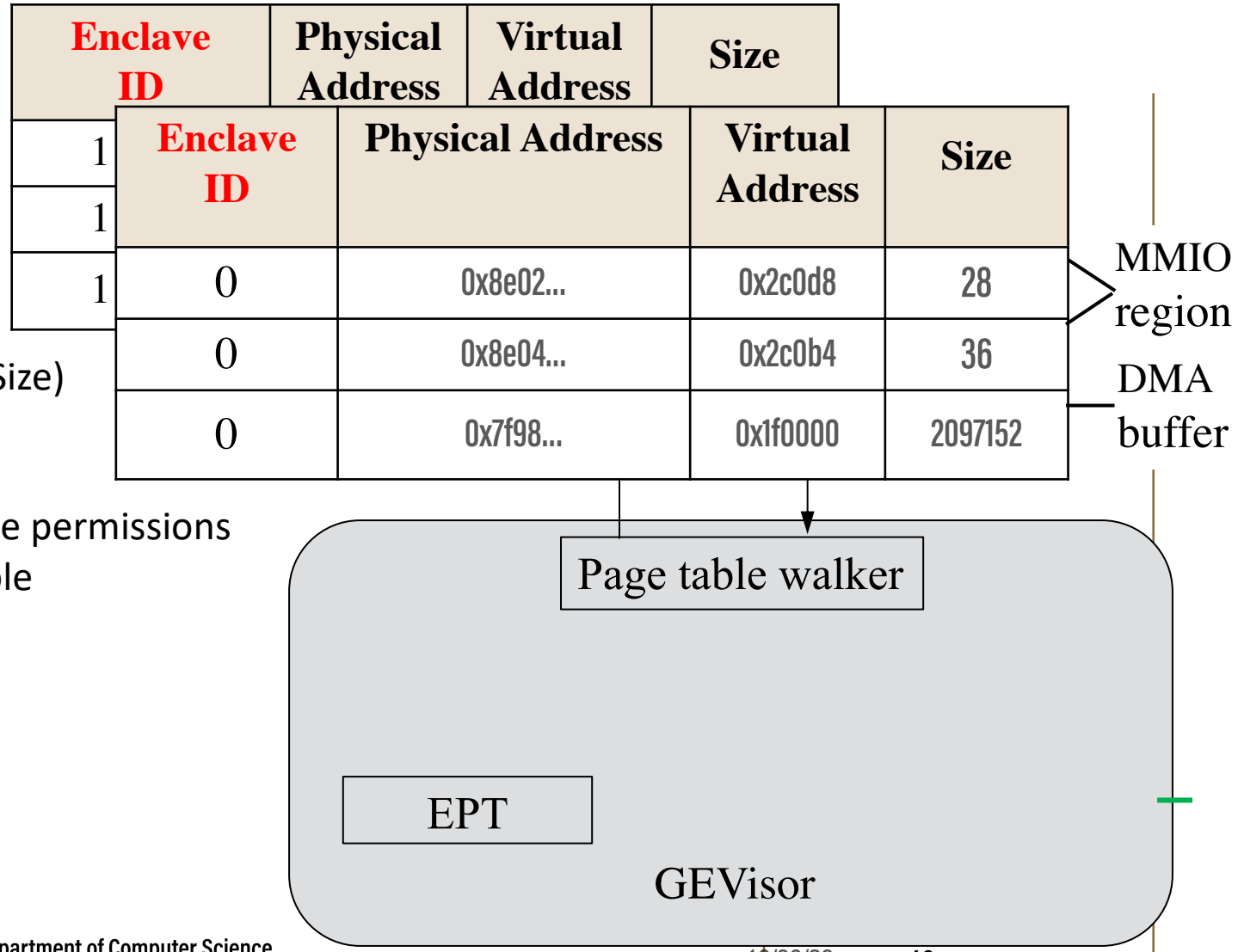
GPU I/O access control

- Reserved Mrtable (ID, PA, VA, Size)
- Pass (ID, VA, Size)
- Filling PA



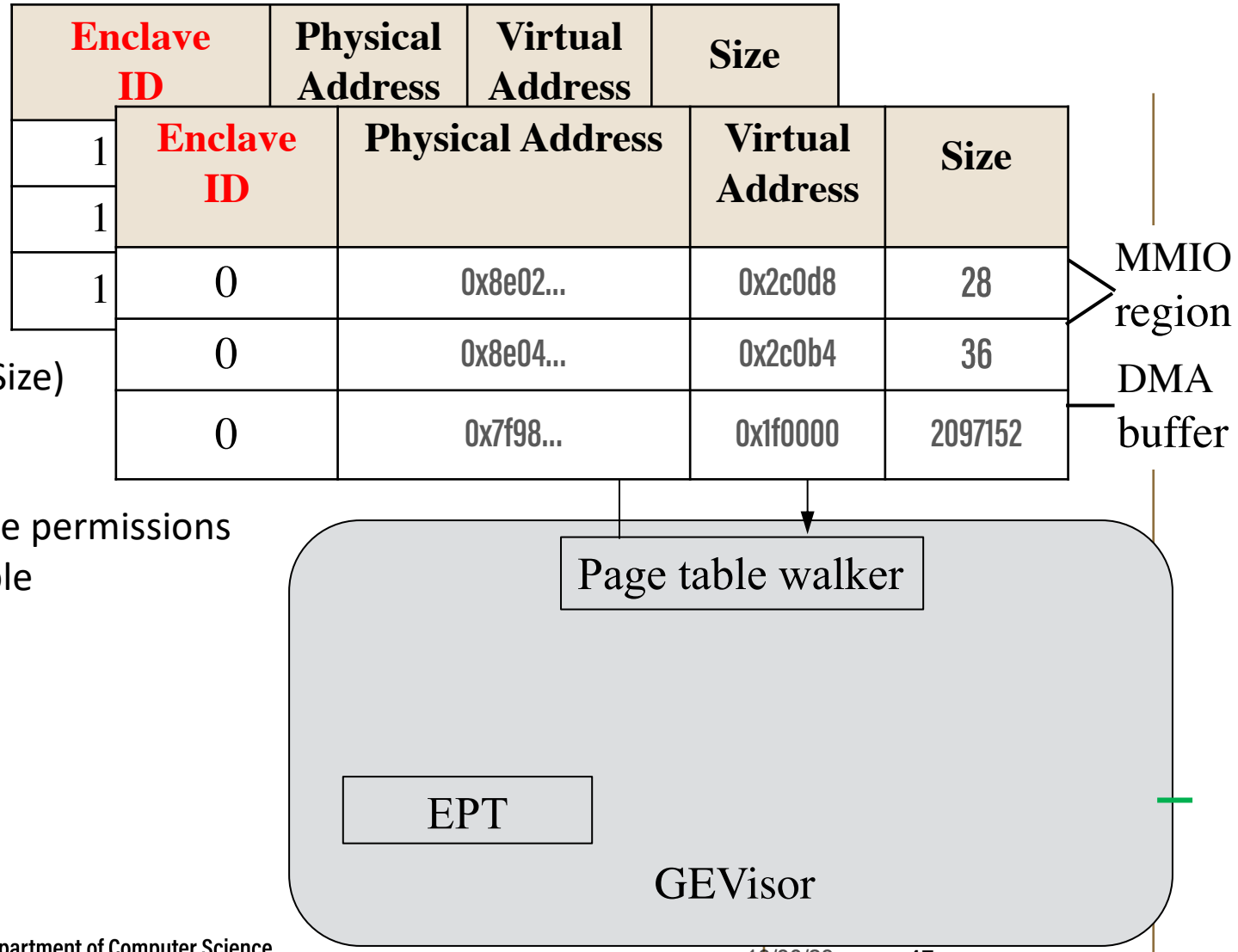
GPU I/O access control

- Reserved Mrtable (ID, PA, VA, Size)
- Pass (ID, VA, Size)
- Filling PA
- EPT: Remove the read and write permissions of each page within the MRtable



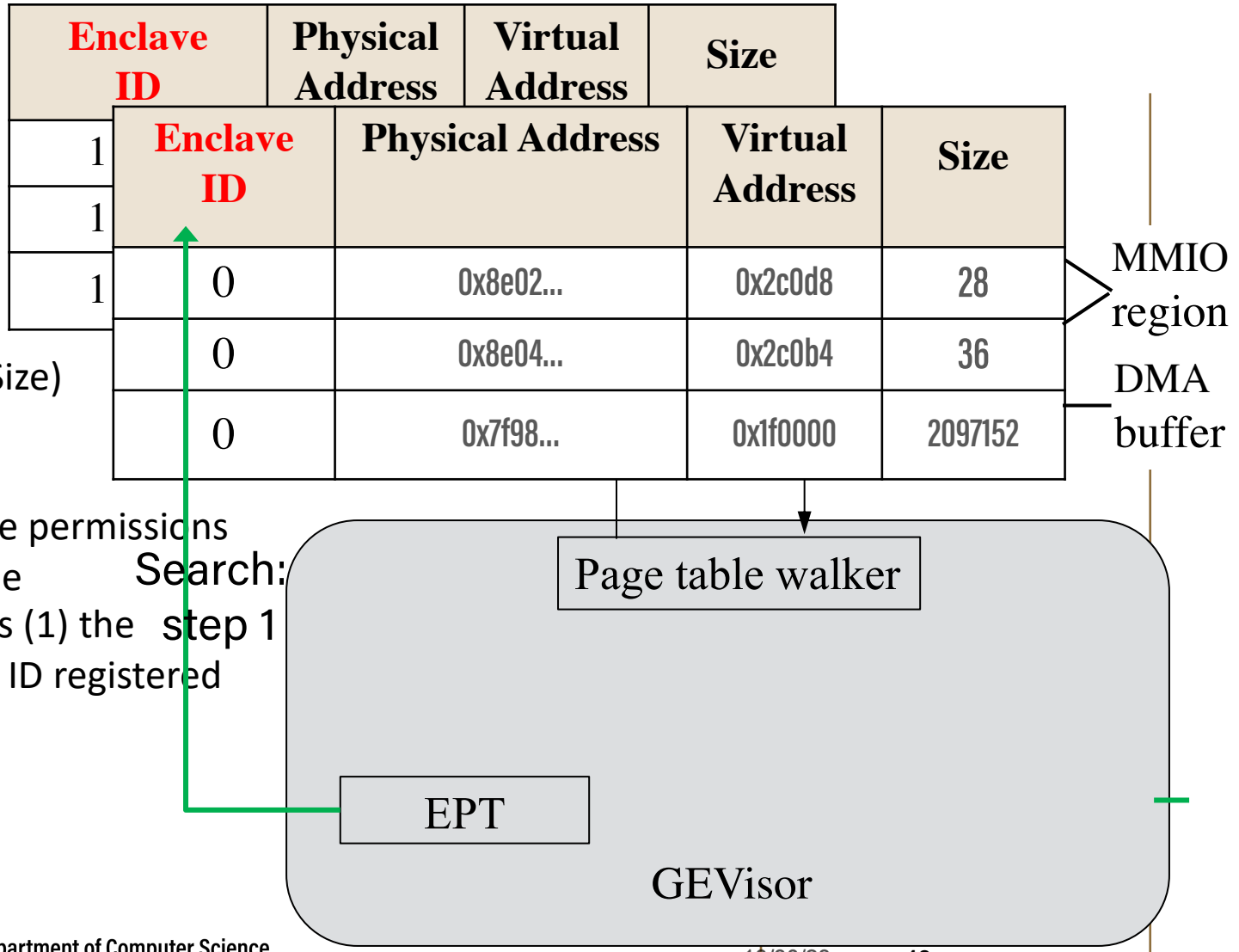
GPU I/O access control

- Reserved Mrtable (ID, PA, VA, Size)
- Pass (ID, VA, Size)
- Filling PA
- EPT: Remove the read and write permissions of each page within the MRtable



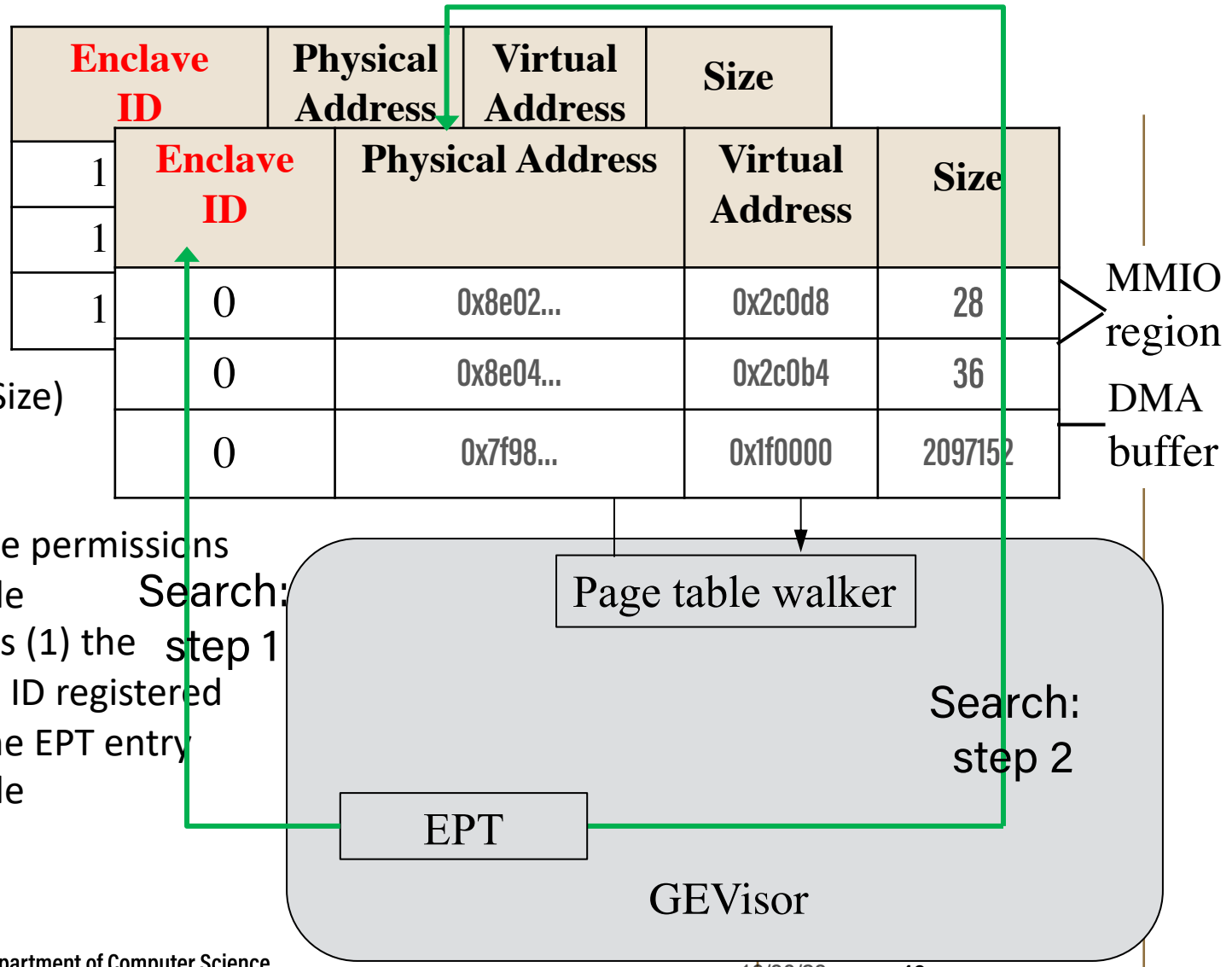
GPU I/O access control

- Reserved Mrtable (ID, PA, VA, Size)
- Pass (ID, VA, Size)
- Filling PA
- EPT: Remove the read and write permissions of each page within the Mrtable
- EPT exception handler validates (1) the current process has an enclave ID registered in the MRtable;



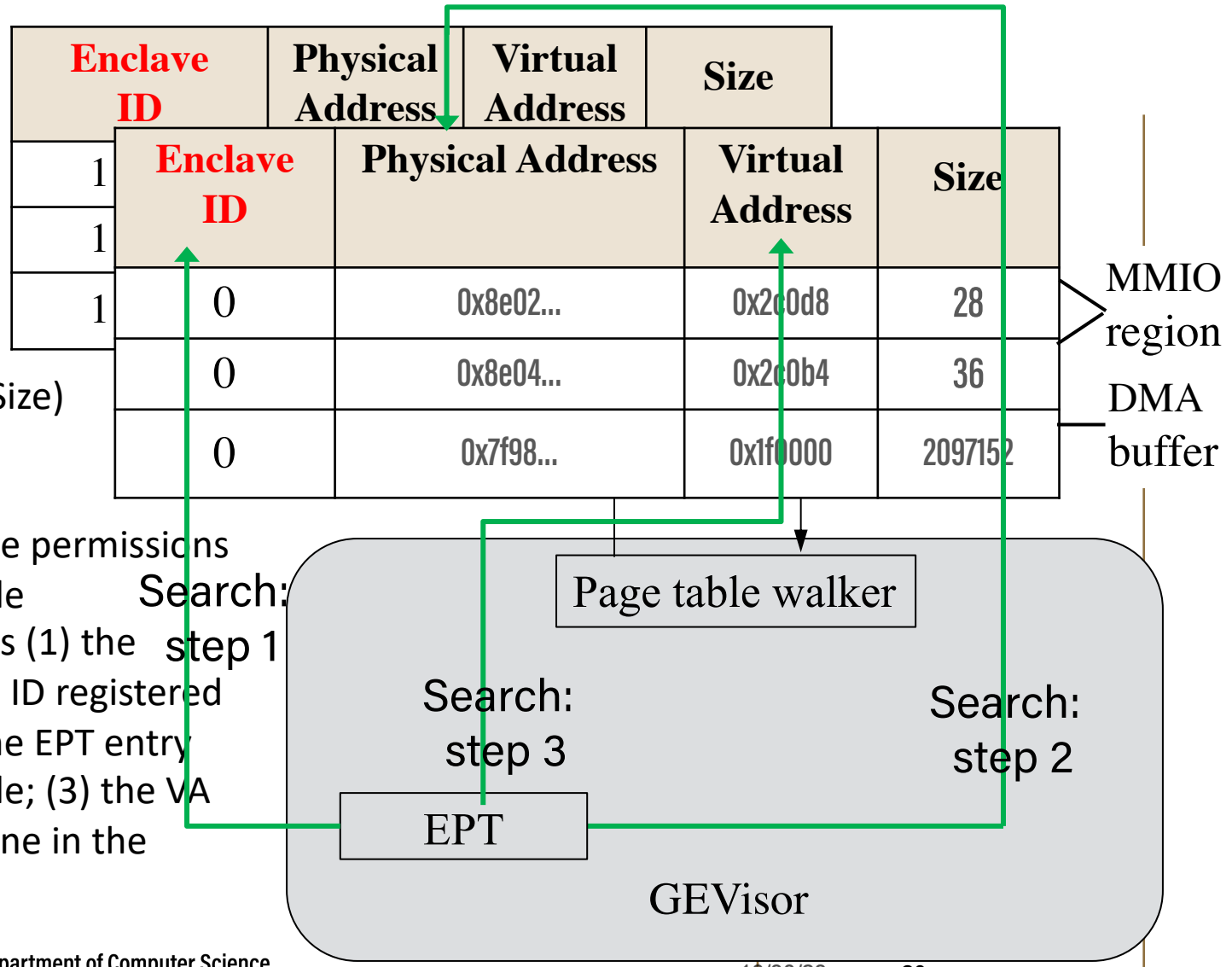
GPU I/O access control

- Reserved Mrtable (ID, PA, VA, Size)
- Pass (ID, VA, Size)
- Filling PA
- EPT: Remove the read and write permissions of each page within the Mrtable
- EPT exception handler validates (1) the current process has an enclave ID registered in the MRtable; (2) the PA in the EPT entry matches the one in the MRtable



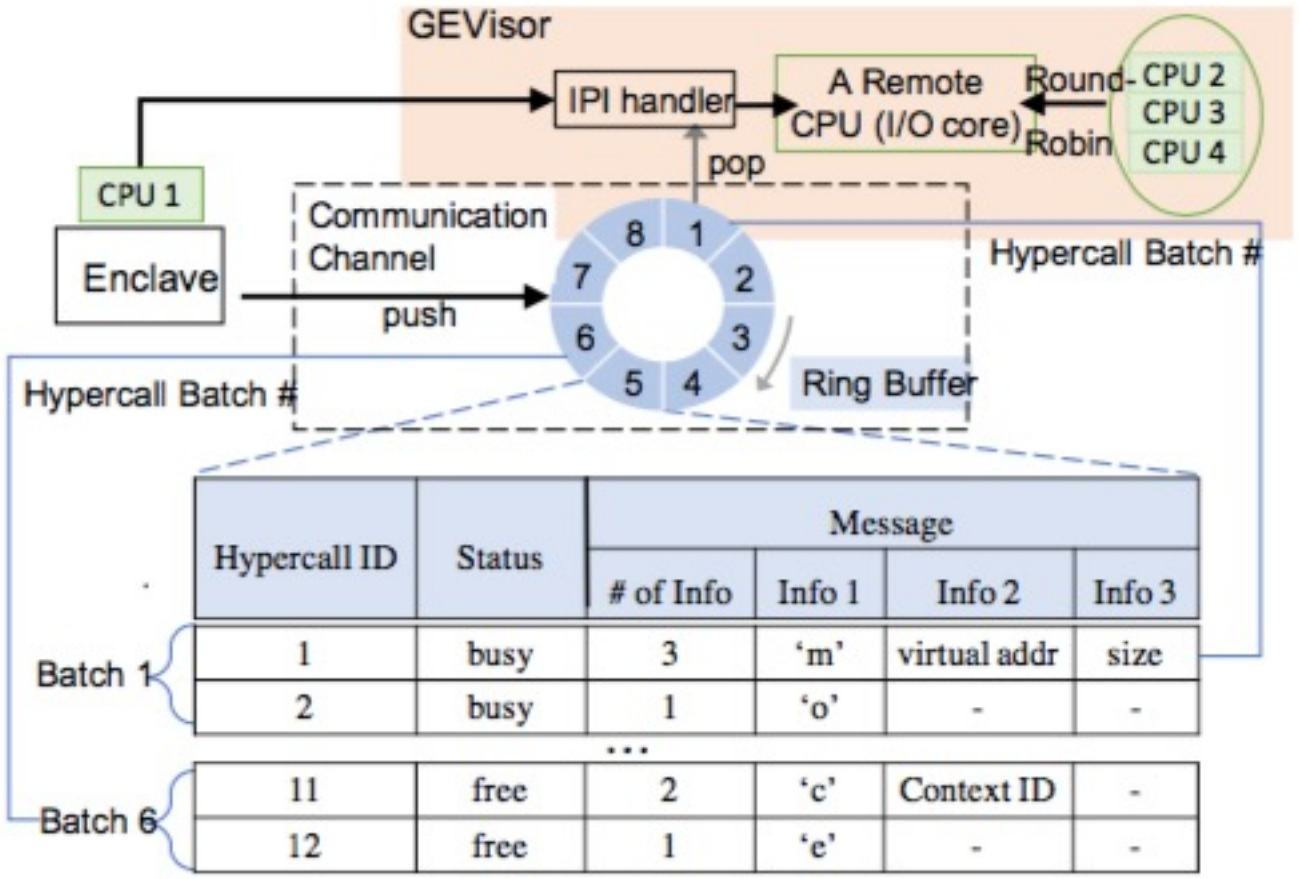
GPU I/O access control

- Reserved Mrtable (ID, PA, VA, Size)
- Pass (ID, VA, Size)
- Filling PA
- EPT: Remove the read and write permissions of each page within the Mrtable
- EPT exception handler validates (1) the current process has an enclave ID registered in the MRtable; (2) the PA in the EPT entry matches the one in the MRtable; (3) the VA in the EPT entry matches the one in the MRtable.



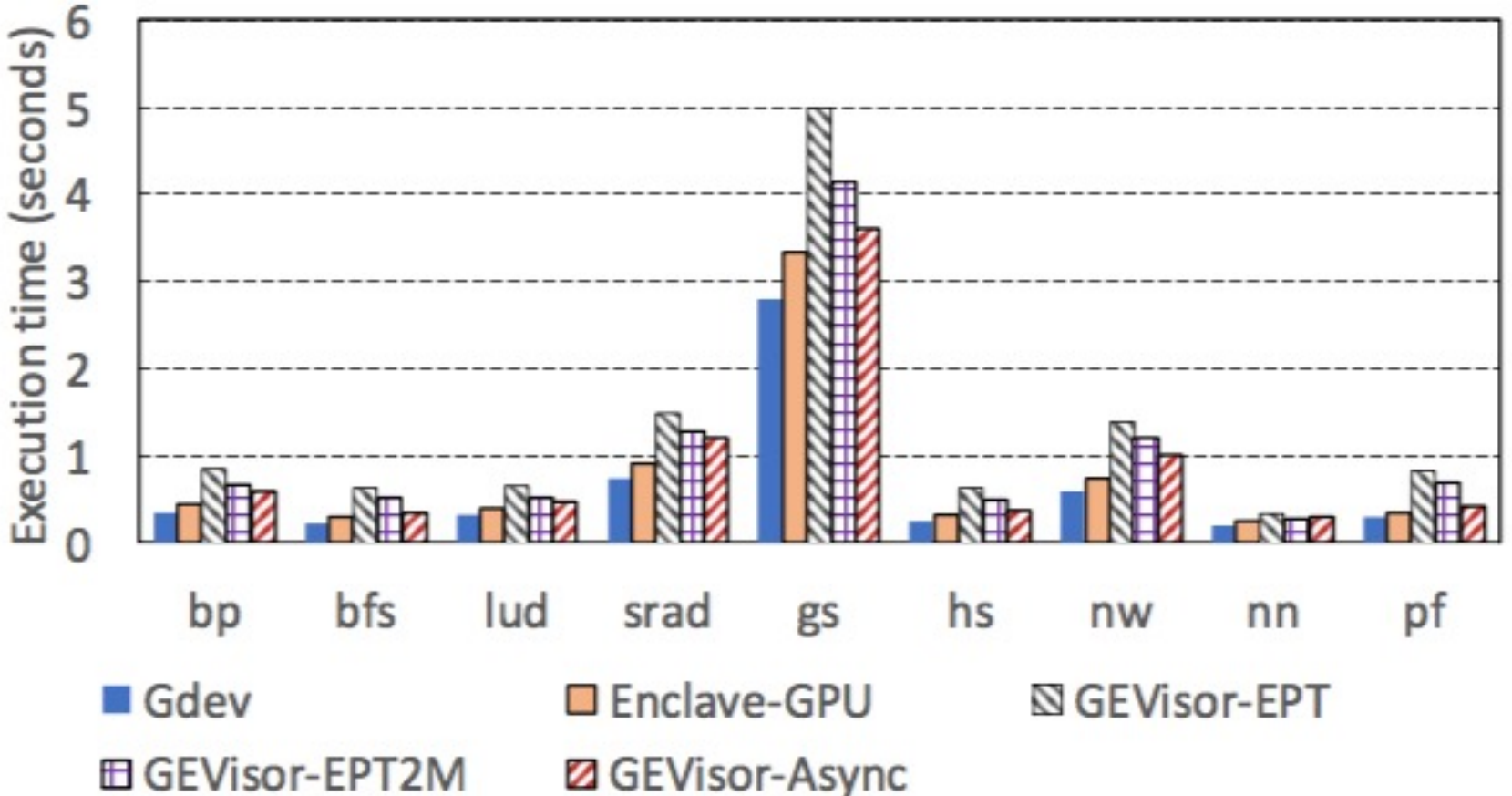
Asynchronous hypercall

- Reserved I/O cores
- Batched hypercall
- Unified hypercall entry format with 6 fields (ID, status, no. of arguments, payload)
- Offload the I/O monitoring task to the remote I/O cores.
- Inter-processor Interrupt (IPI) handler processes the hypercall offloading following a round- robin fashion



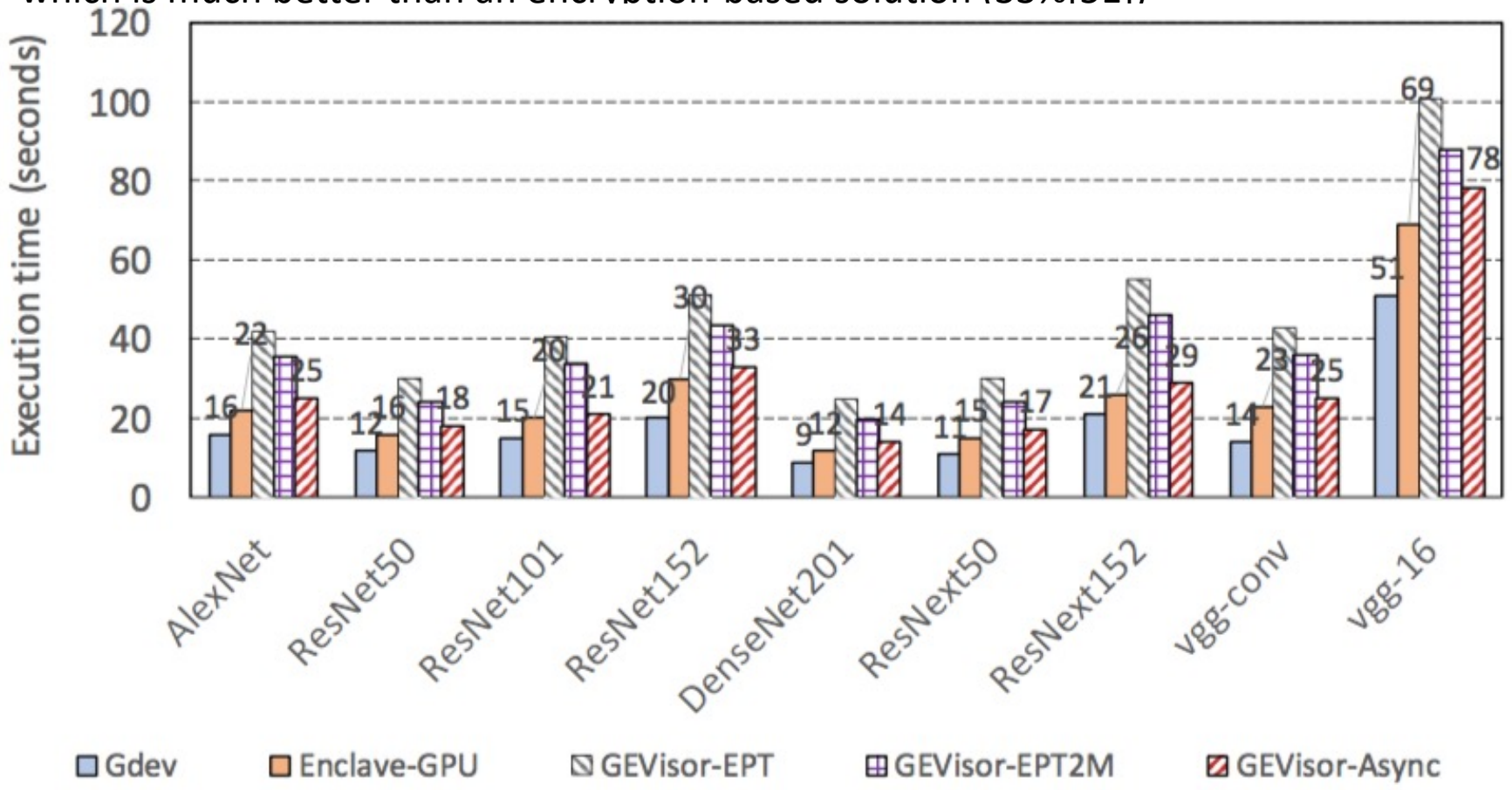
Evaluation (Rodinia benchmarks)

- For I/O-bound workload (bp, bfs, srad, nw, and pf), asynchronous hypercall improves multi-core utilization significantly as the message batch size increases dramatically.
- Computation-bound workload, hs, lud, and especially nn, with small GPU kernels does not benefit from asynchronous hypercall. (small amount of hypercall requests does not amortize the IPI overhead)

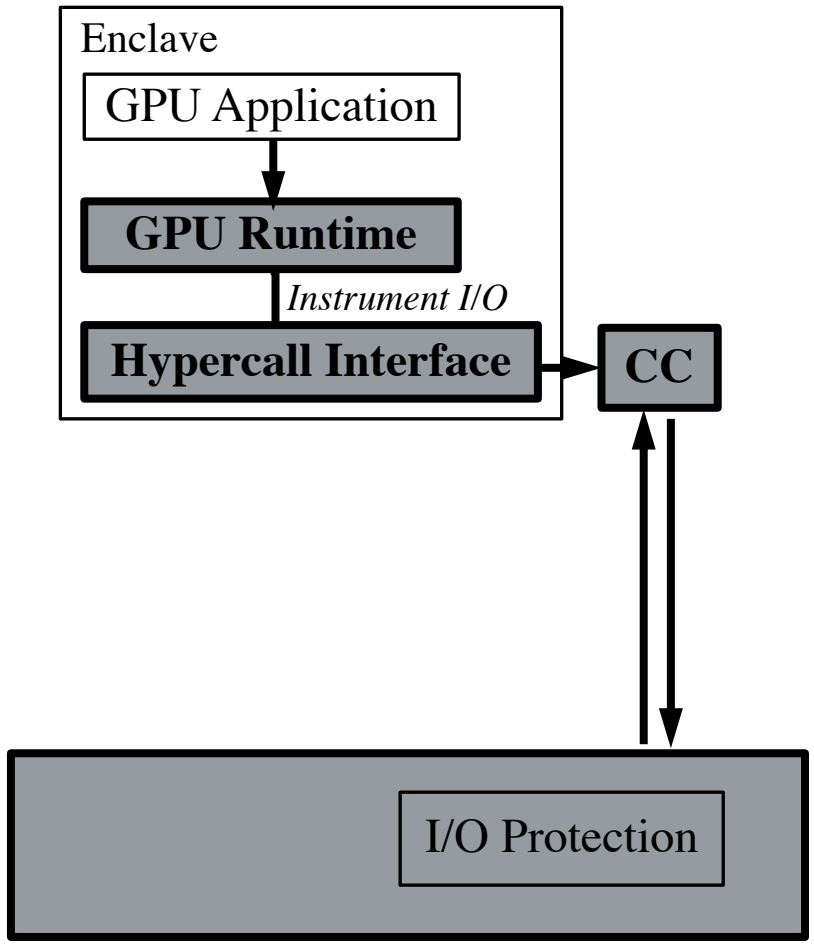


Evaluation (Darknet benchmarks)

- Asynchronous hypercall (GEVISOR-Async) has only a 13.1% overhead on average, which is much better than an encryption-based solution (33%[51])



Takeaway from this talk!



GEVisor

THANK YOU!
Q&A