IPADS INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEMS

USENIX ATC '24

CPC: Flexible, Secure, and Efficient CVM Maintenance with Confidential Procedure Calls

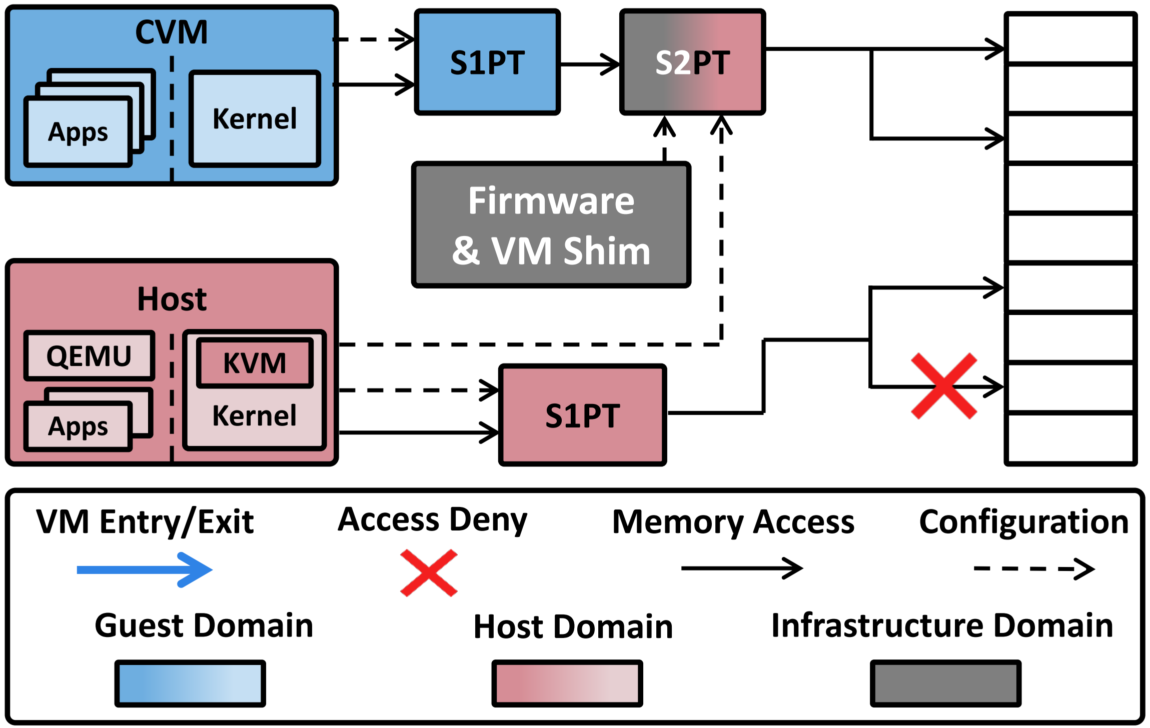
Jiahao Chen, Zeyu Mi, Yubin Xia, Haibing Guan, Haibo Chen

Shanghai Jiao Tong University

饮 水 思 源 • 爱 国 荣 校

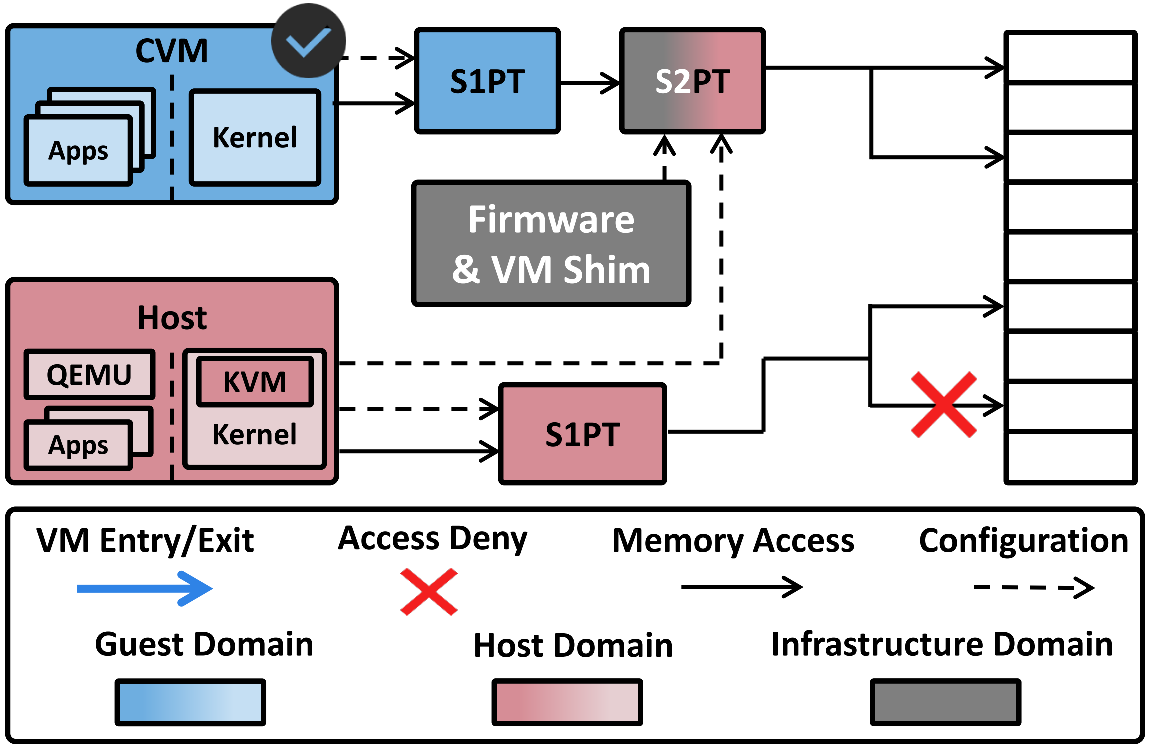
CVMs Safeguard Tenants' Privacy

- CVM—Run VMs in TEE



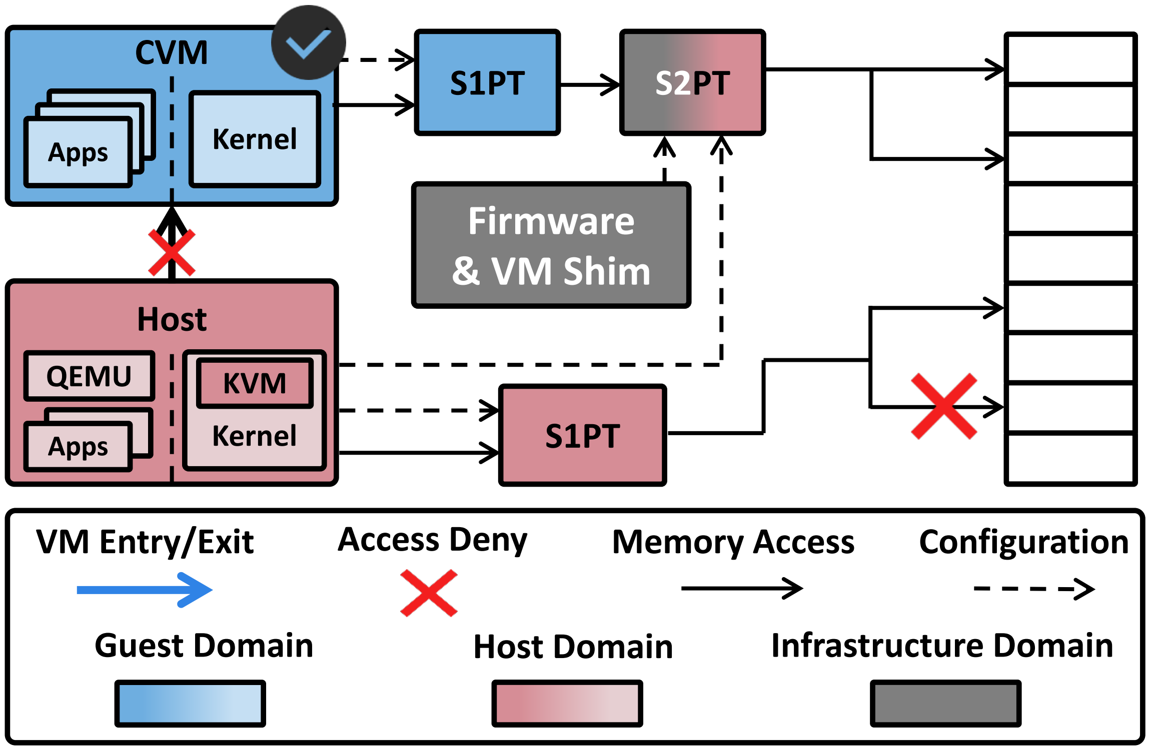
CVMs Safeguard Tenants' Privacy

- CVM—Run VMs in TEE
  - Image attestation



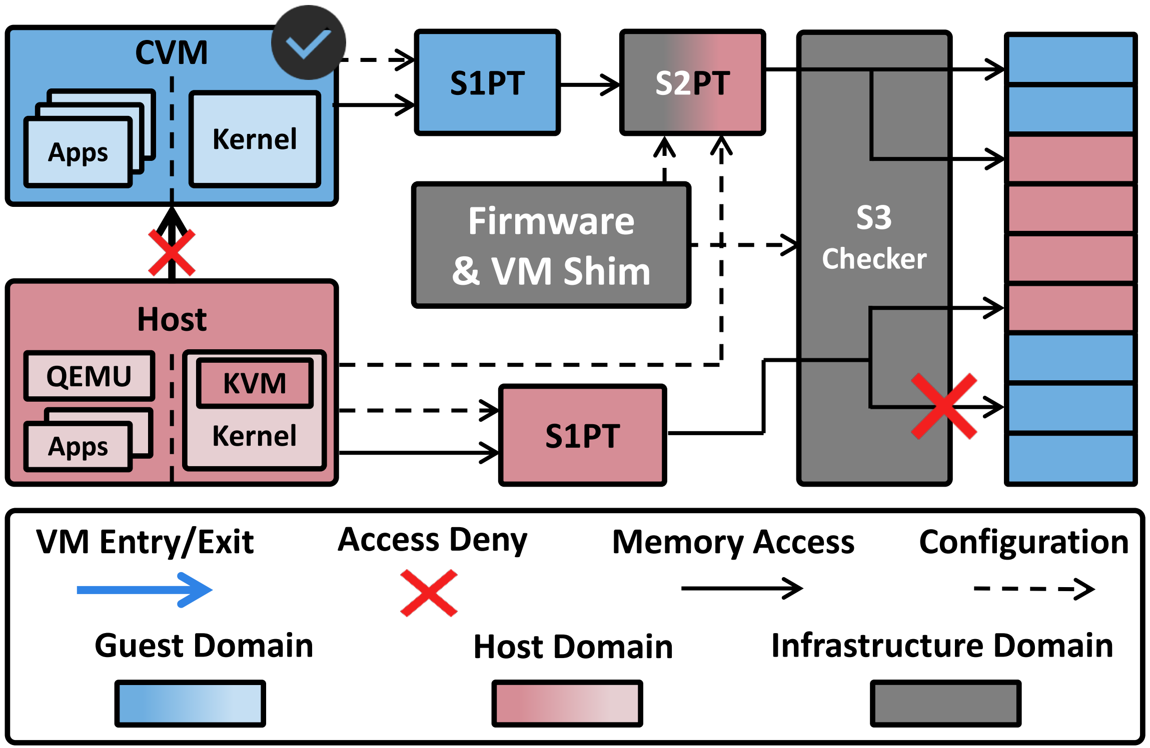
CVMs Safeguard Tenants' Privacy

- CVM—Run VMs in TEE
  - Image attestation
  - Register states can not be accessed by the host



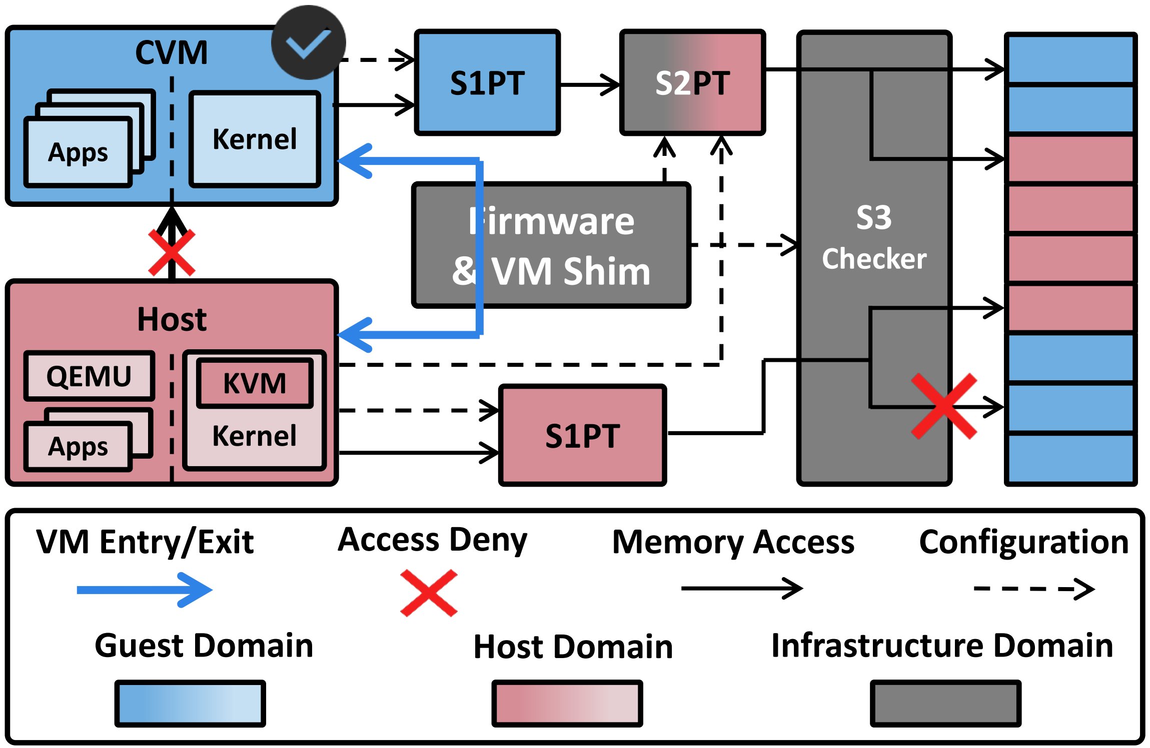
CVMs Safeguard Tenants' Privacy

- CVM—Run VMs in TEE
  - Image attestation
  - Register states can not be accessed by the host
  - Stage-3 memory protection for guests' private memory



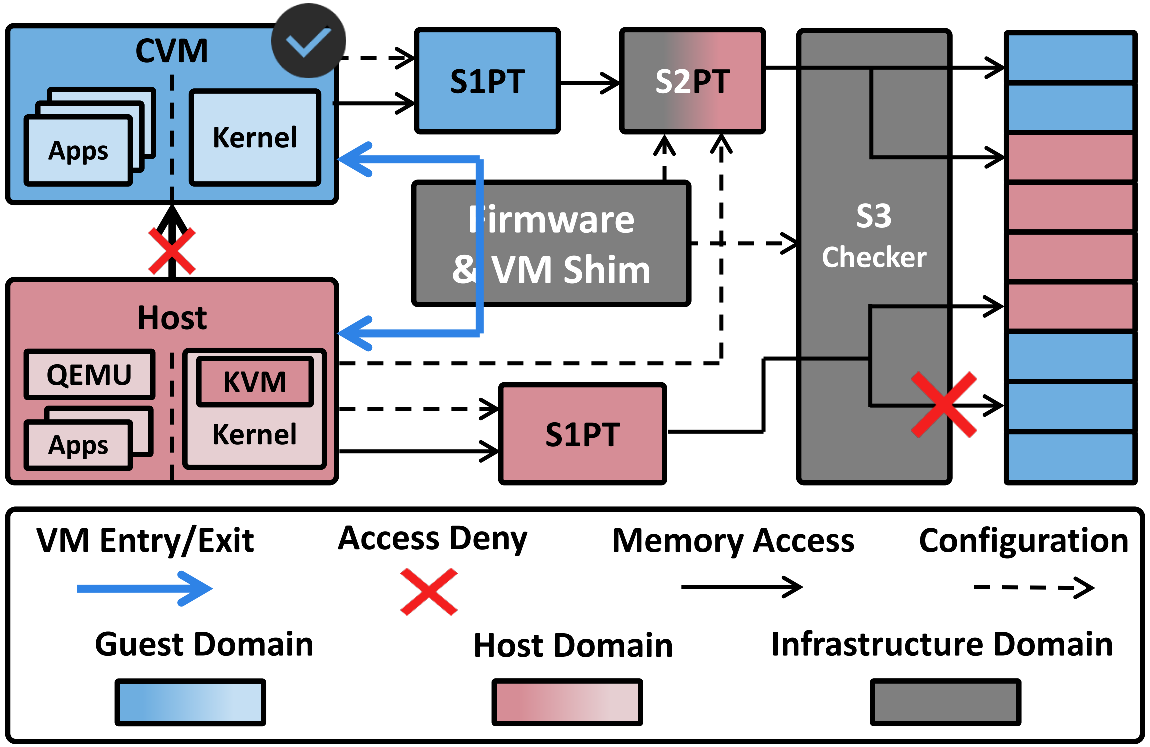
CVMs Safeguard Tenants' Privacy

- CVM—Run VMs in TEE
  - Image attestation
  - Register states can not be accessed by the host
  - Stage-3 memory protection for guests' private memory
  - VM exits are filtered by the trusted FW or shim



CVMs Safeguard Tenants' Privacy

- CVM—Run VMs in TEE
  - Image attestation
  - Register states can not be accessed by the host
  - Stage-3 memory protection for guests' private memory
  - VM exits are filtered by the trusted FW or shim



AMD SEV, Intel TDX, ARM CCA, and RISC-V CoVE

Nightmare for Maintenance

Alibaba Cloud

Azure

Host OS

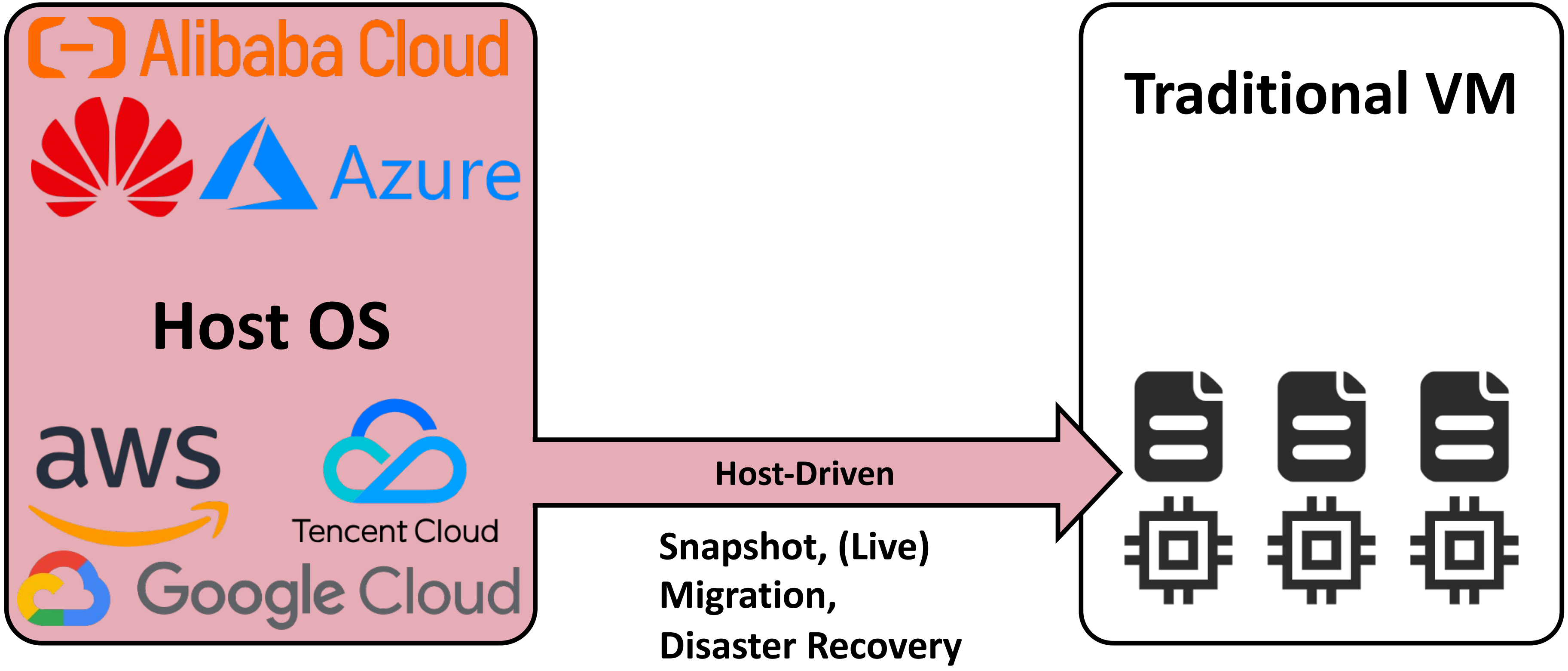
aws

Tencent Cloud

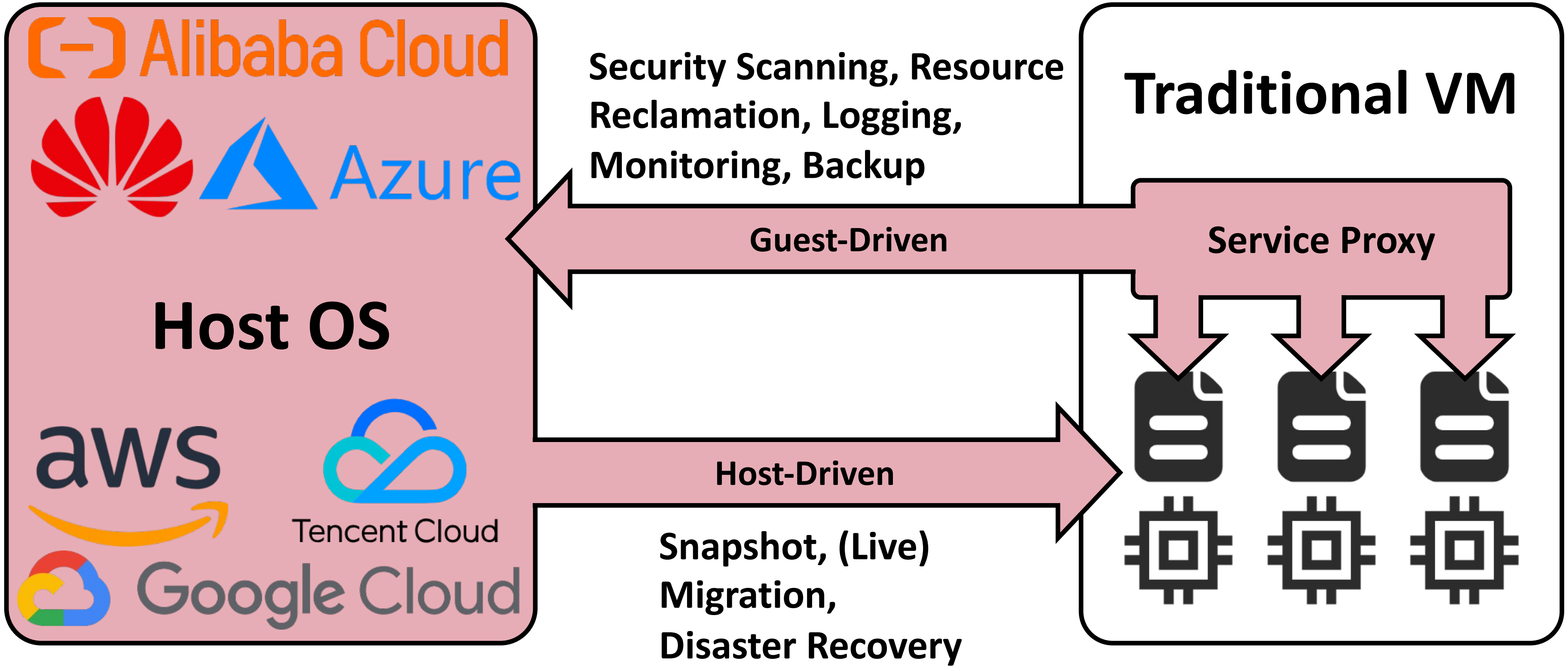
Google Cloud

Traditional VM

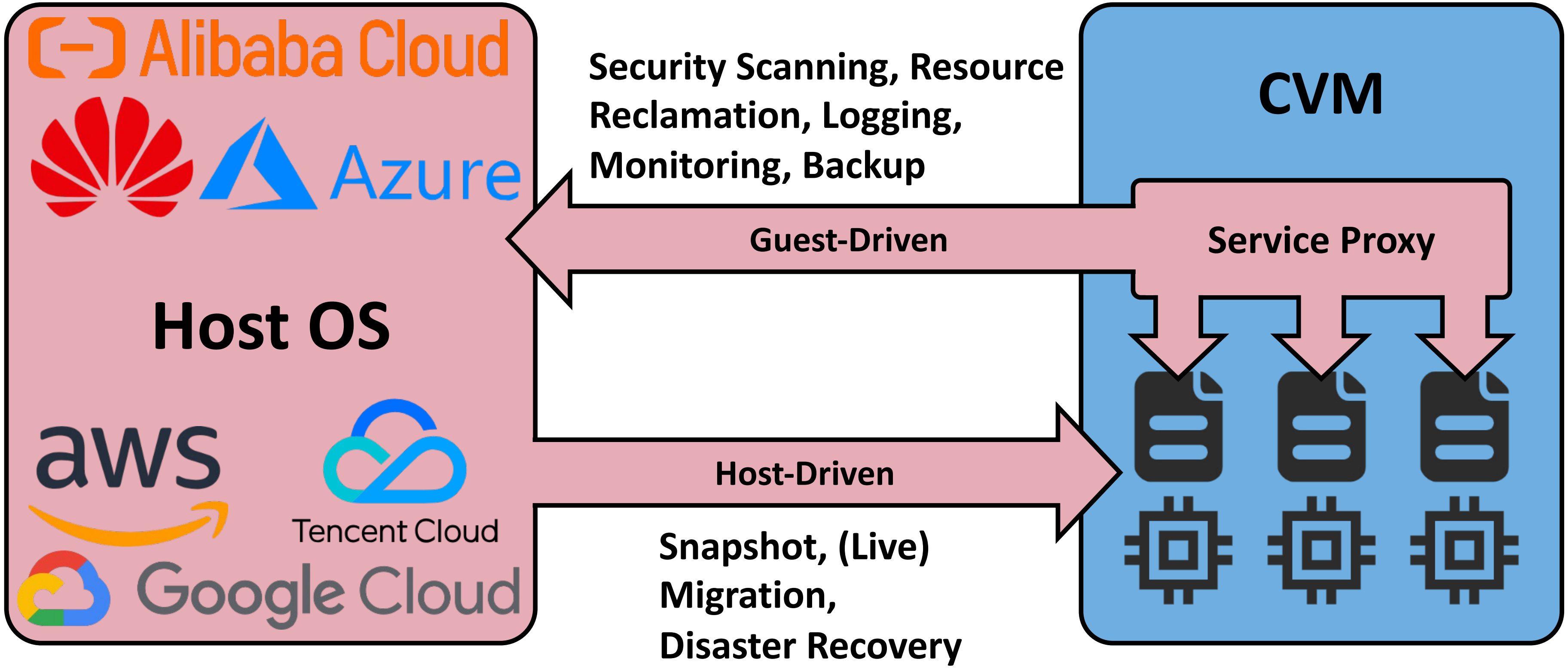
Nightmare for Maintenance



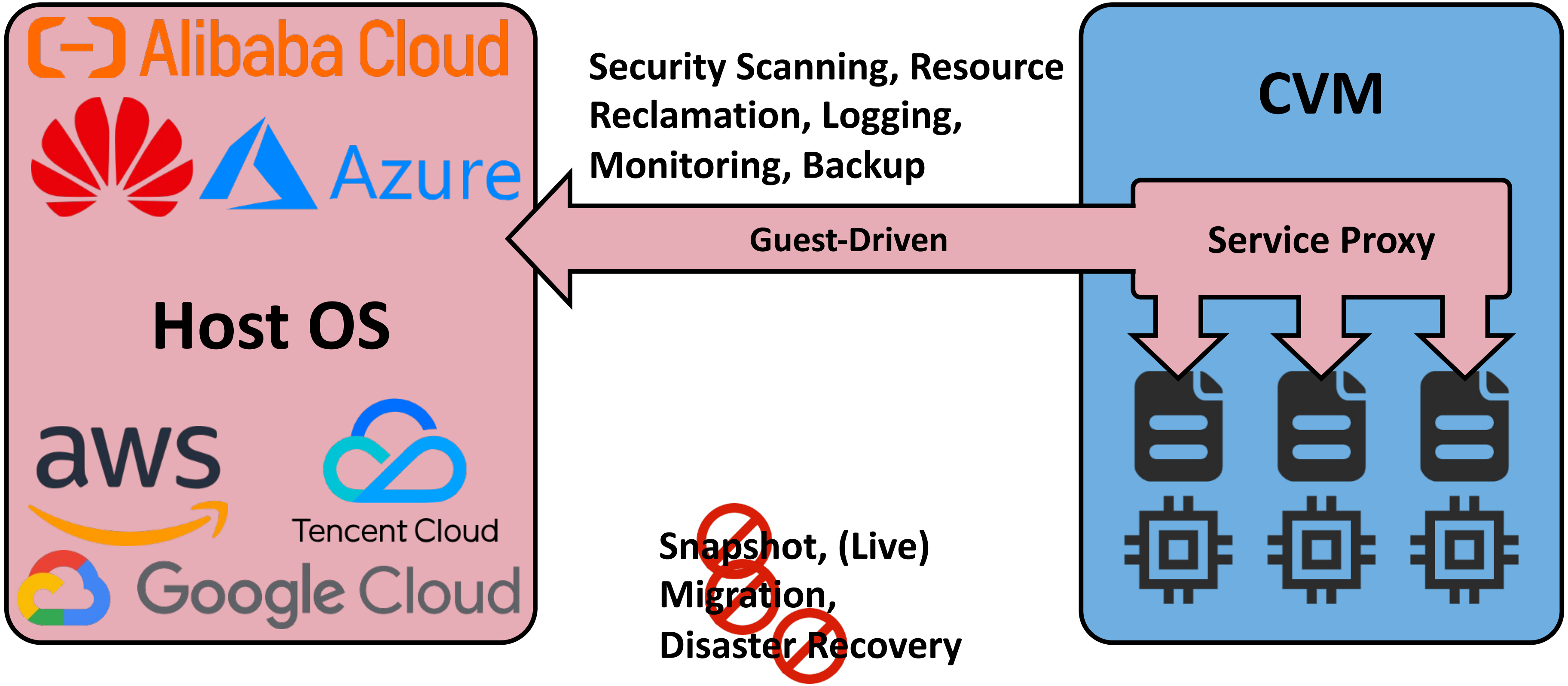
Nightmare for Maintenance



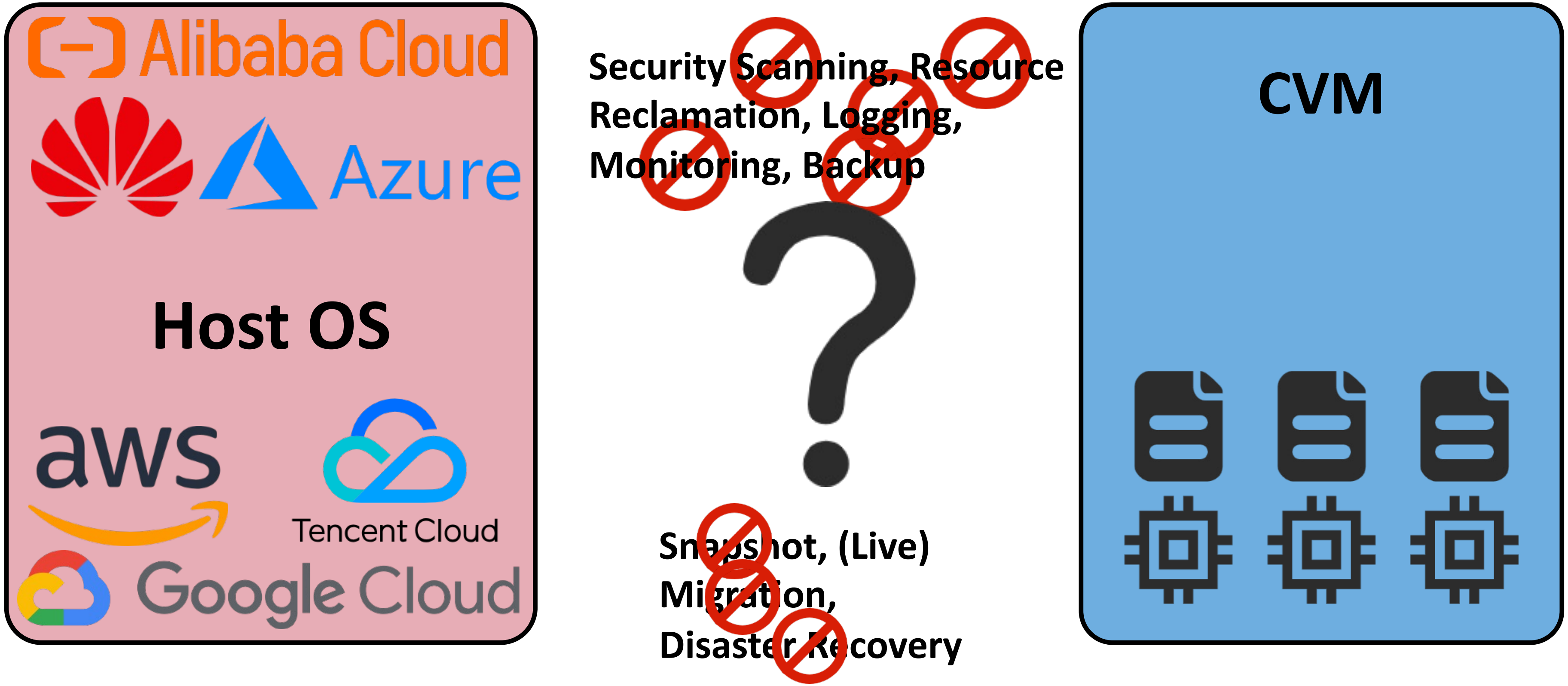
Nightmare for Maintenance



Nightmare for Maintenance

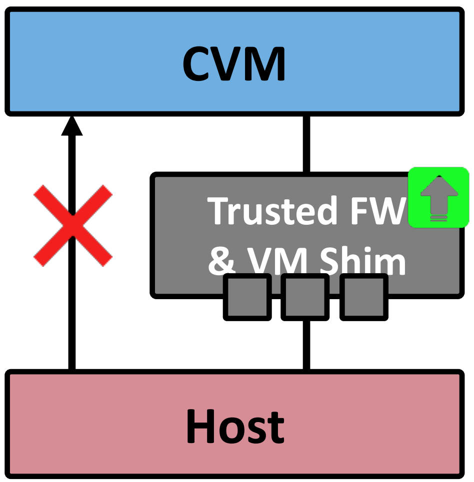


Nightmare for Maintenance



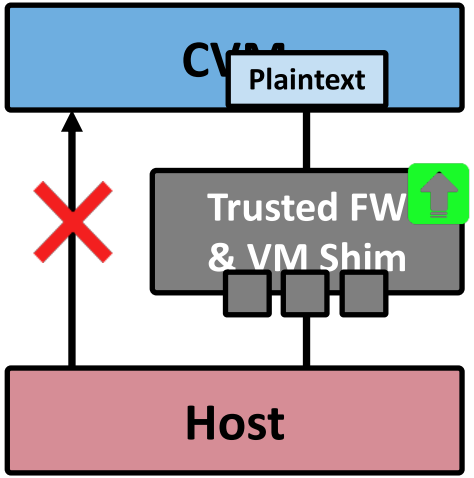
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host



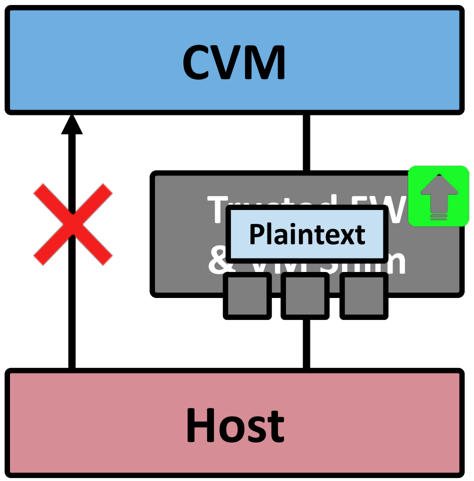
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption



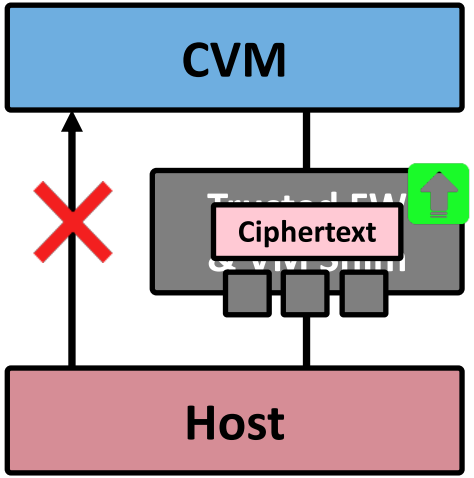
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption



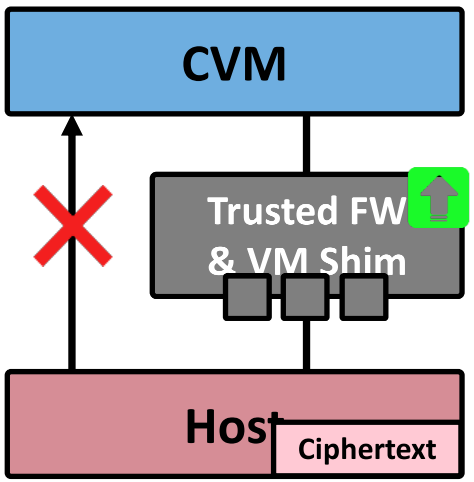
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption



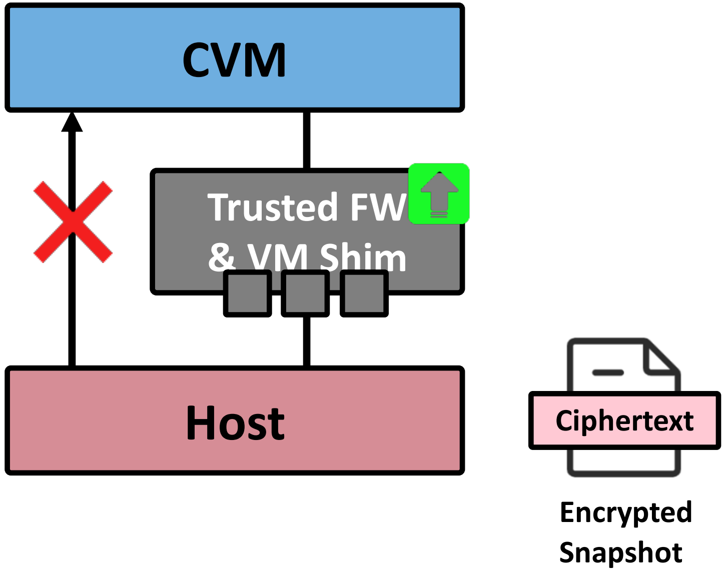
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption



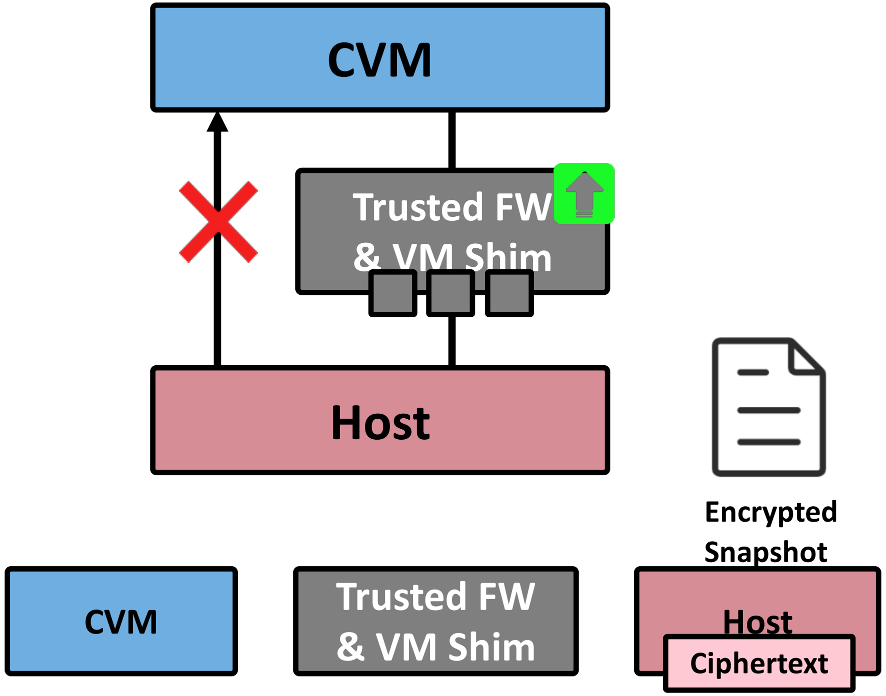
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption
  - Ciphertext is transferred by the host
  - Snapshot, (Live) Migration



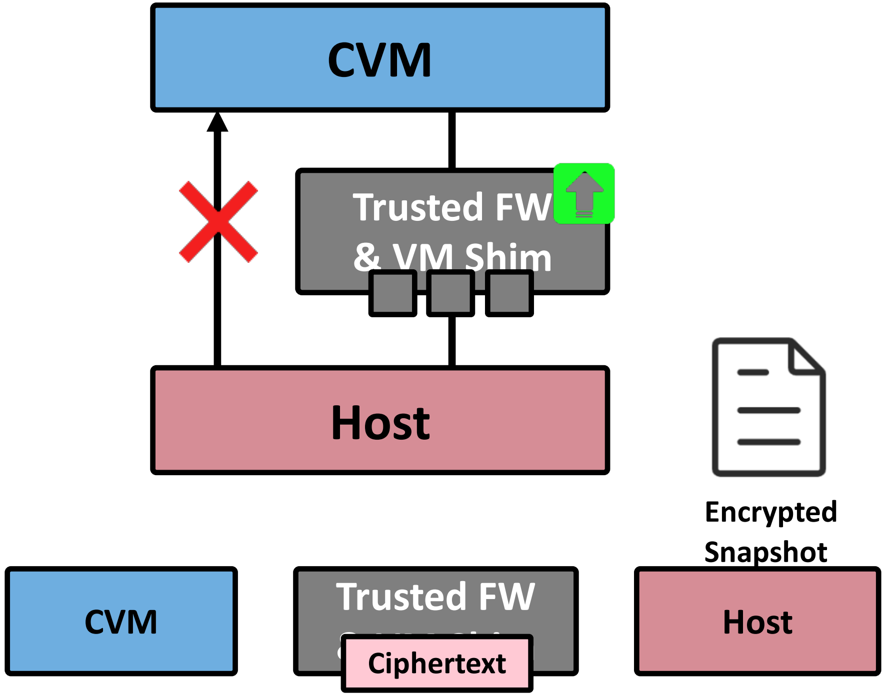
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption
  - Ciphertext is transferred by the host
  - Snapshot, (Live) Migration



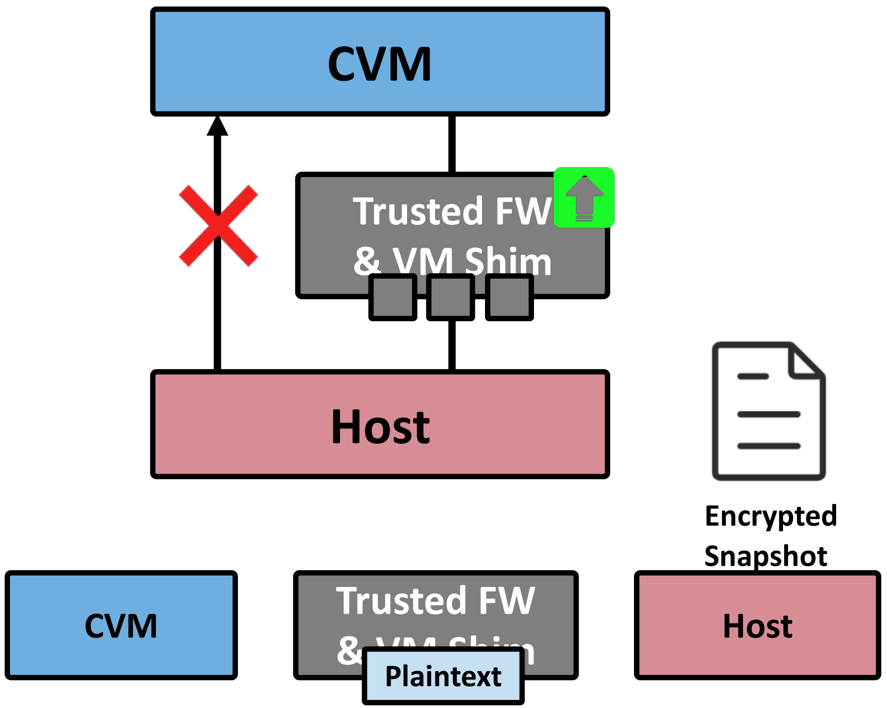
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption
  - Ciphertext is transferred by the host
  - Snapshot, (Live) Migration



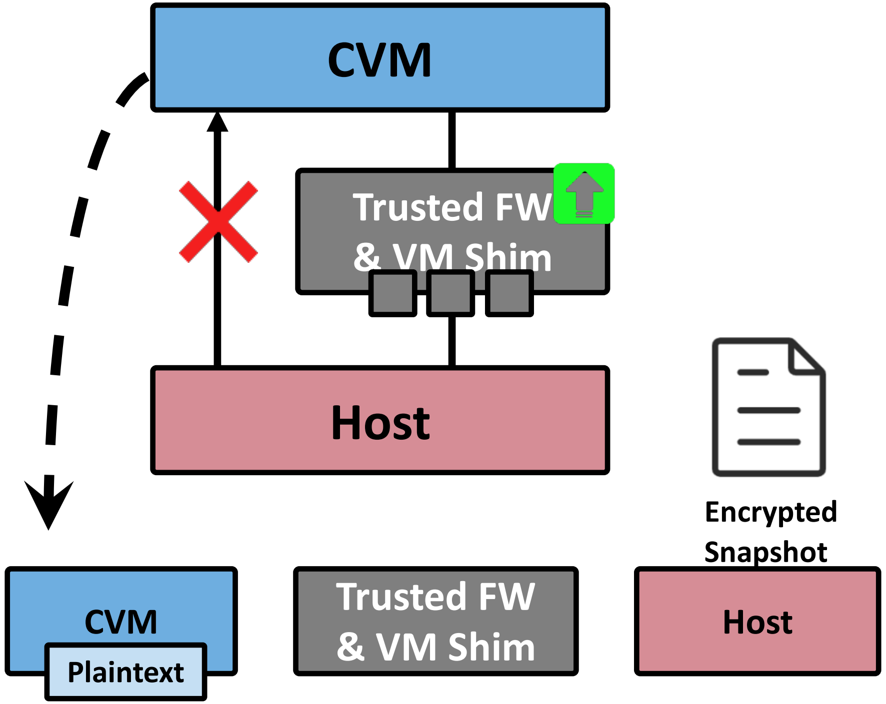
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption
  - Ciphertext is transferred by the host
  - Snapshot, (Live) Migration



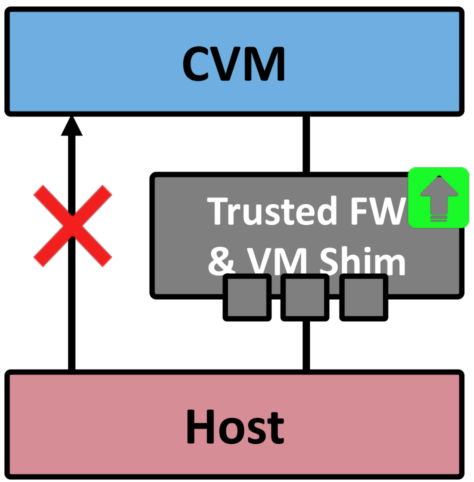
Salvage Relying on Hardware Vendors

- Upgrade the trusted firmware and export new interfaces to the host
  - Extracting the private memory and states with encryption
  - Inserting the private memory and states with decryption
  - Ciphertext is transferred by the host
  - Snapshot, (Live) Migration



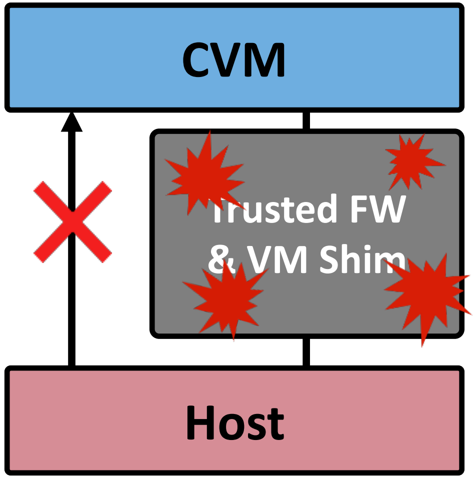
Limitations of Existing Approaches

- Inflexibility
  - Slow updates for FW/HW
  - Reboot the machine
  - Lack of cross-platform compatibility



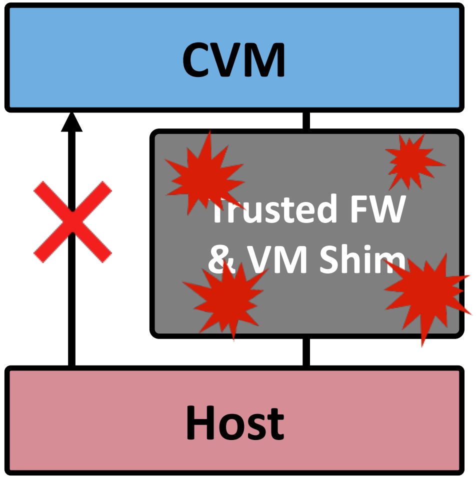
Limitations of Existing Approaches

- Inflexibility
  - Slow updates for FW/HW
  - Reboot the machine
  - Lack of cross-platform compatibility
- Security degradation
  - Inflated Trusted Firmware
  - Universal TCB for the entire system



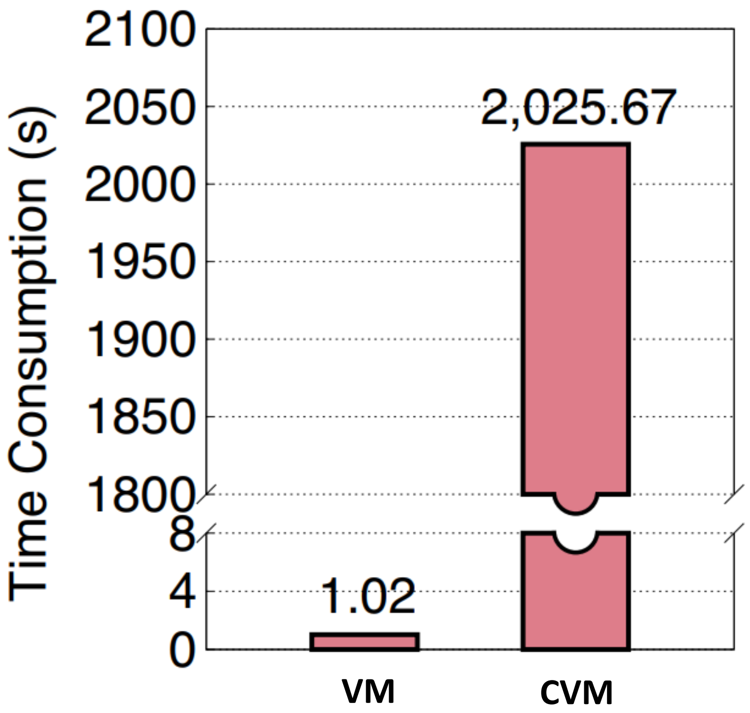
Limitations of Existing Approaches

- Inflexibility
  - Slow updates for FW/HW
  - Reboot the machine
  - Lack of cross-platform compatibility
- Security degradation
  - Inflated Trusted Firmware
  - Universal TCB for the entire system
- Poor Performance



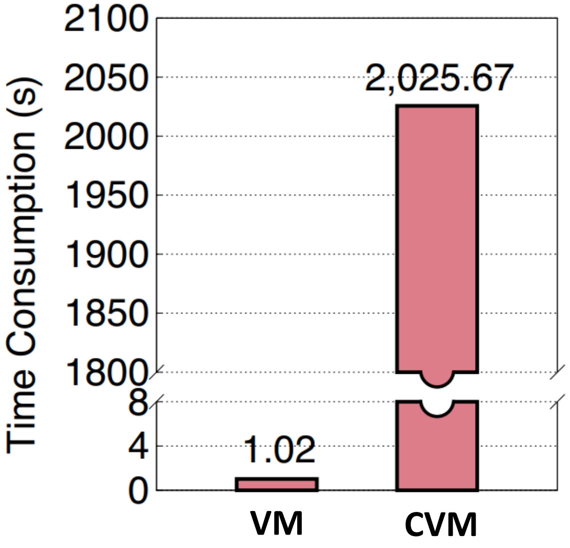
Limitations of Existing Approaches

- AMD SEV official live migration solution takes **1986x slower**?
  - Traditional VM -- 1.02s
  - Confidential VM -- 2,025.67s
- Testbed Configuration
  - An AMD platform with 128 cores
  - VMs with a vCPU & 2GB DRAM
  - SRC & DST VM run on the same machine to minimize the impact of unstable networks

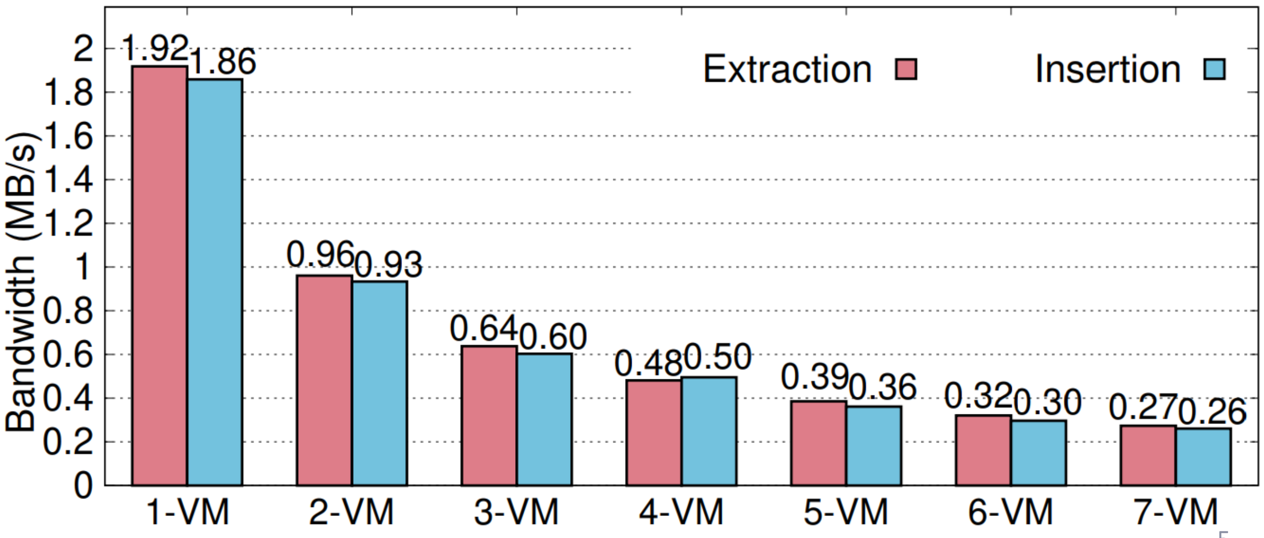


Limitations of Existing Approaches

- AMD SEV official live migration solution takes **1986x slower**?
  - Traditional VM -- 1.02s
  - Confidential VM -- 2,025.67s



- The trusted firmware runs on the AMD-SP
  - 32-bit ARM core, limited computing power, **1.92MB/s**
  - Shared out by all CVMs



Goals

Flexibility

- Enable cloud vendors and tenants to customize and update the maintenance modules
- Updates without having to suspend/migration VMs or reboot the machine
- Compatible with all major CVM platforms without hardware modifications

Goals

Flexibility

- Enable cloud vendors and tenants to customize and update the maintenance modules
- Updates without having to suspend/migration VMs or reboot the machine
- Compatible with all major CVM platforms without hardware modifications

Security

- Uphold the security of current CVMs
- Maintain the clear security boundary between the guest and host

Goals

Flexibility

- Enable cloud vendors and tenants to customize and update the maintenance modules
- Updates without having to suspend/migration VMs or reboot the machine
- Compatible with all major CVM platforms without hardware modifications

Security

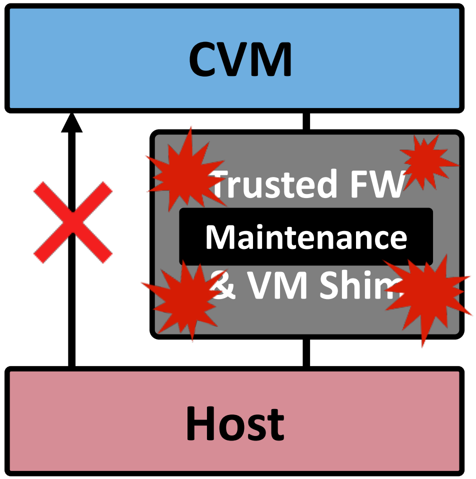
- Uphold the security of current CVMs
- Maintain the clear security boundary between the guest and host

Efficiency

- Mitigate the performance limitations caused by factors such as guest workloads & AMD-SPs

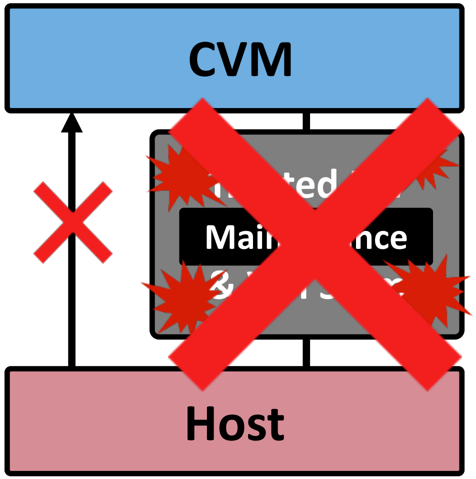
Root Cause

Inappropriate choice of vantage point for maintenance modules

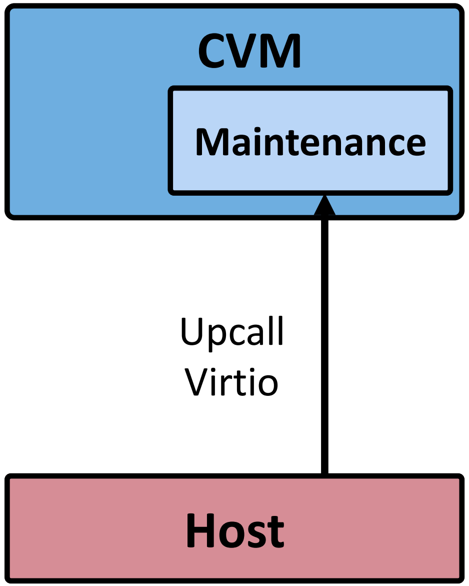


Root Cause

Inappropriate choice of vantage point for maintenance modules

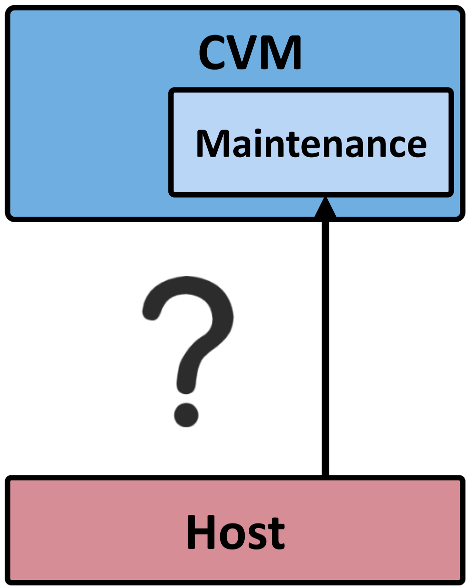


Better Vantage Point



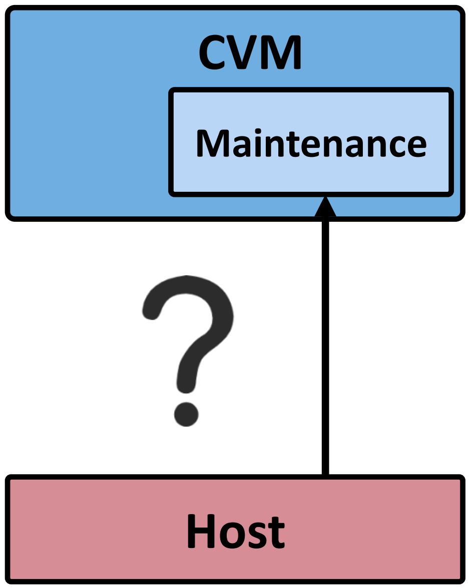
New Solution

- Performance degradation due to resource contention with the guest workload
  - **3x** slowdown in resource reclamation scenarios



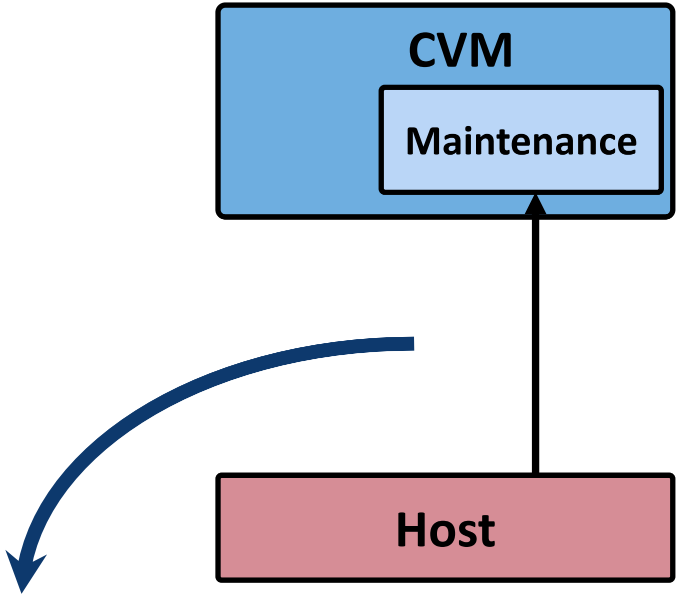
New Solution

- Performance degradation due to resource contention with the guest workload
  - **3x** slowdown in resource reclamation scenarios
- No fault tolerance
  - Several operations require to work correctly even when the guest OS fails
    - Disaster recovery, monitoring



New Solution

- Performance degradation due to resource contention with the guest workload
  - **3x** slowdown in resource reclamation scenarios
- No fault tolerance
  - Several operations require to work correctly even when the guest OS fails
    - Disaster recovery, monitoring
- A new mechanism capable of providing the host with the semantics of:



Host invocation of targeted maintenance operations with **SEPARATE** and **PROTECTED** resources.

Observation & Key Idea

- CVMs limit the host's intrusive access to the guest's **data plane**
  - The hypervisor still exerts influence over the **control plane**
  - E.g., scheduling vCPUs

Observation & Key Idea

- CVMs limit the host's intrusive access to the guest's **data plane**
  - The hypervisor still exerts influence over the **control plane**
  - E.g., scheduling vCPUs

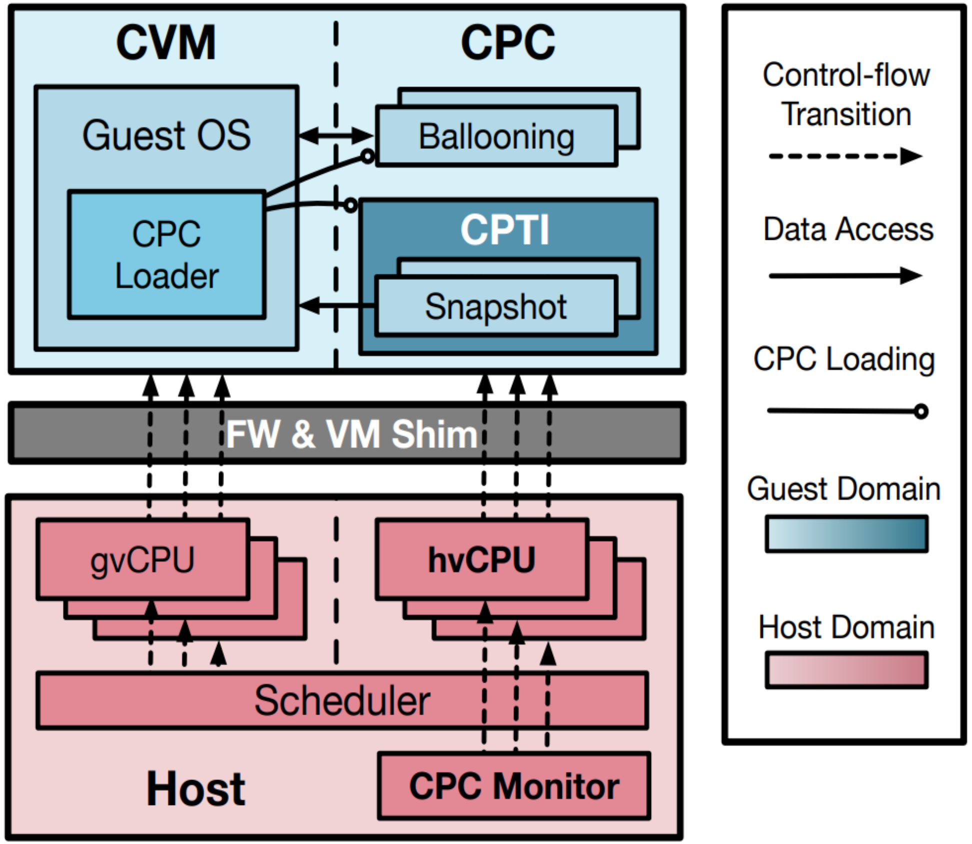
Extend the semantics of ***vCPU scheduling*** into the semantics of ***host invocations of maintenance procedures***

CPC

Confidential Procedure Calls

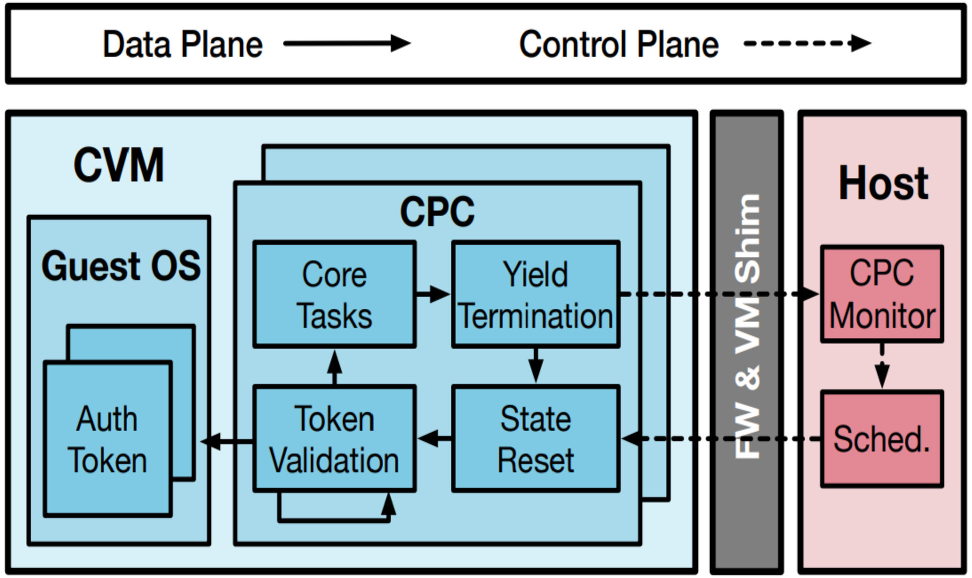
Confidential Procedure Calls

- Add extra vCPUs to the CVMs
  - **hvCPUs:** vCPUs for the host
  - hvCPUs do not participate in the standard host kernel scheduling
  - hvCPUs are bound with maintenance modules
  - Host OS awakens the hvCPU thread according to the maintenance scenarios
- Guest OS runs on normal vCPUs
  - **gvCPUs:** vCPUs for the guest



CPC State Machine

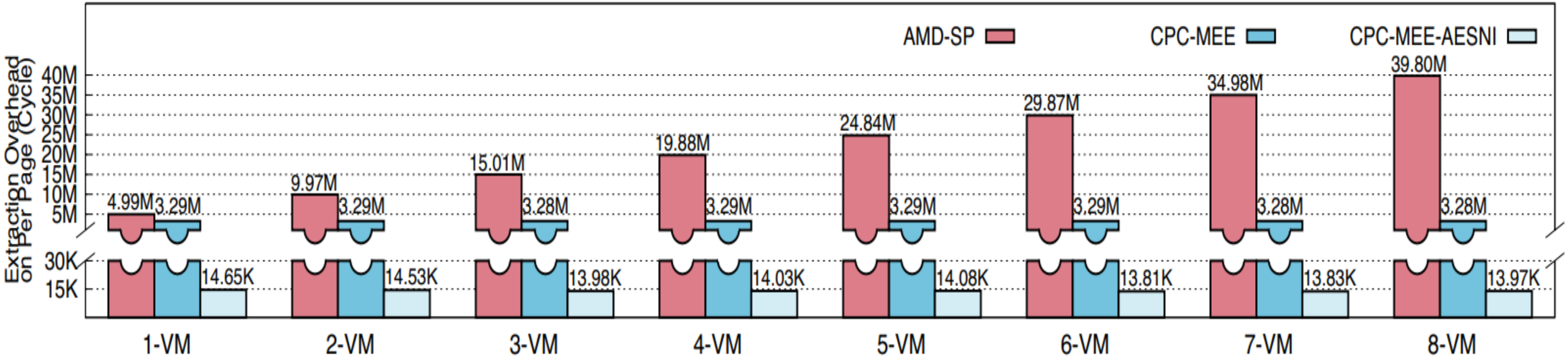
- A state machine driven by both the **in-host control plane** and the **in-guest data plane**
  - The procedure works when the host OS awakens its hvCPU thread
  - Authorization tokens to prevent overcalls
  - Infinite loop, can be called multiple times



- Maintain the **clear security boundary** between the guest and host
- Reuse current **mature mechanisms** and **simple interfaces**

Performance of CPC-Snapshot

- CPC-Snapshot via simple SW:
  - 1-VM: **34% faster**
  - 8-VM: **12x faster**, more VMs, more improvement
  - Good scalability
- CPC-Snapshot with AESNI:
  - 1-VM: **341x faster**
  - 8-VM: **2849x faster**, more VMs, more improvement
  - Still excellent scalability

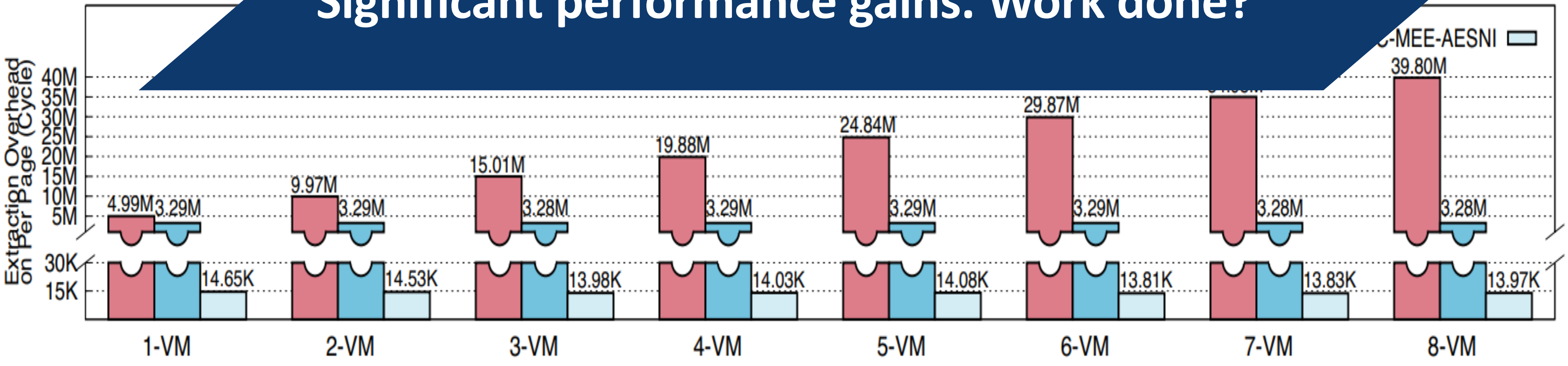


Performance of CPC-Snapshot

- CPC-Snapshot via simple SW:
  - 1-VM: **34% faster**
  - 8-VM: **12x faster**, more VMs, more improvement
  - Go

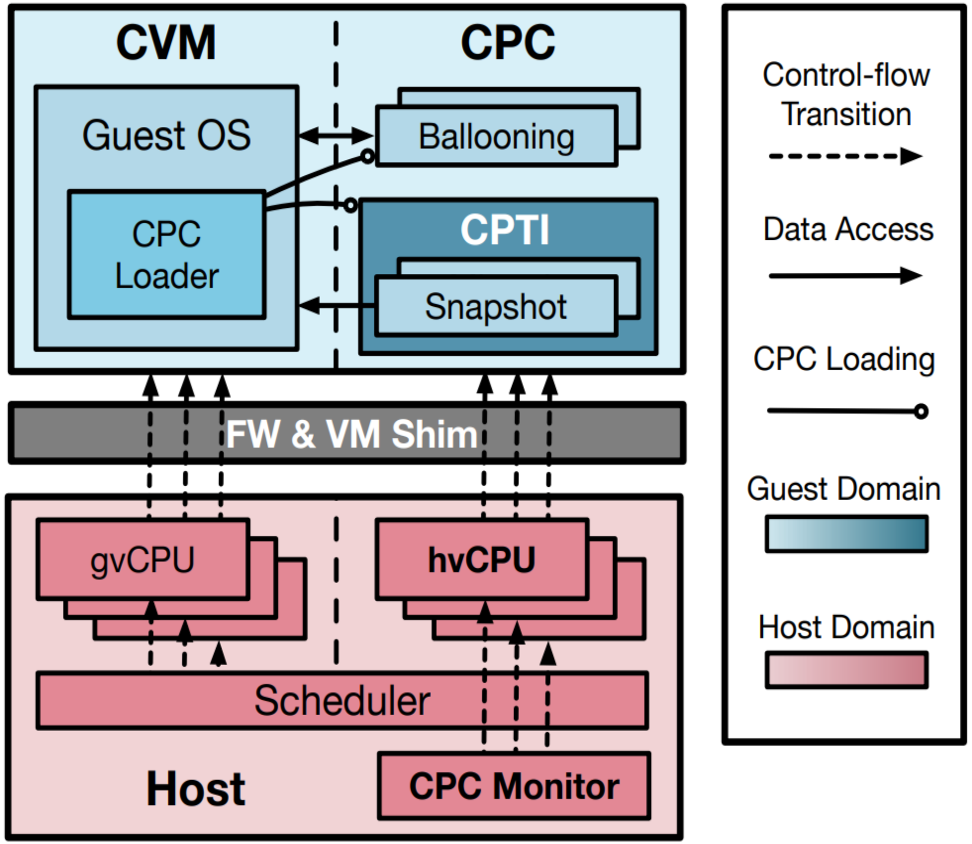
- CPC-Snapshot with AESNI:
  - 1-VM: **341x faster**
  - 8-VM: **2849x faster**, more VMs, more improvement

Significant performance gains. Work done?



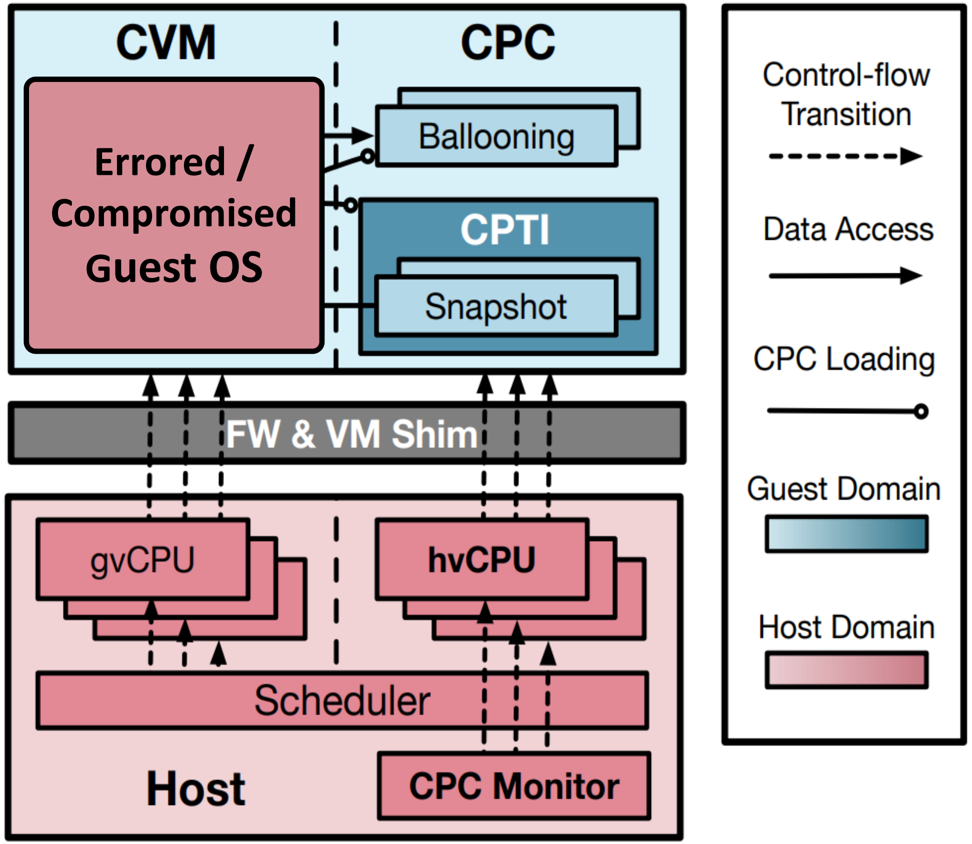
Tougher threat models than CVMs

- Part of maintenance modules should work correctly even when the guest OS errors
  - E.g., disaster recovery, data backup, error diagnose



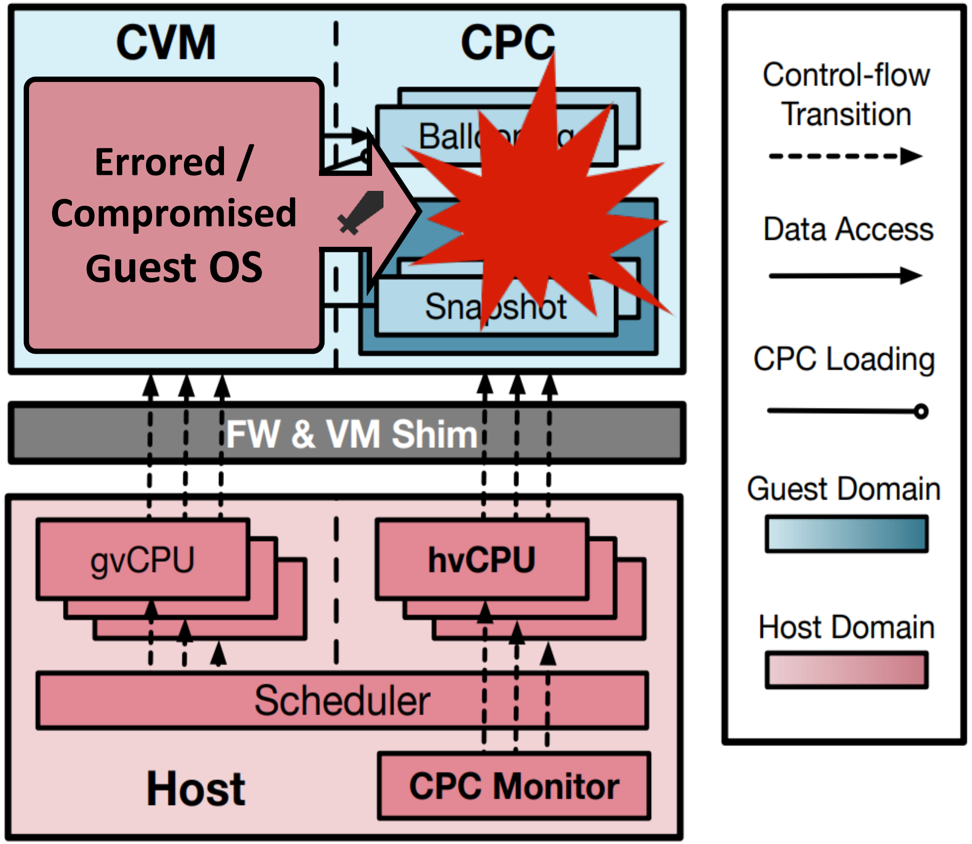
Tougher threat models than CVMs

- Part of maintenance modules should work correctly even when the guest OS errors
  - E.g., disaster recovery, data backup, error diagnose
- Huge TCB from the guest OS
  - Usually Linux
  - Vulnerable, crash-prone, insecure



Tougher threat models than CVMs

- Part of maintenance modules should work correctly even when the guest OS errors
  - E.g., disaster recovery, data backup, error diagnose
- Huge TCB from the guest OS
  - Usually Linux
  - Vulnerable, crash-prone, insecure
- Current CPC cannot survive tampering with a faulty guest OS
  - Resulting in damage to the CPCs, important data cannot be salvaged

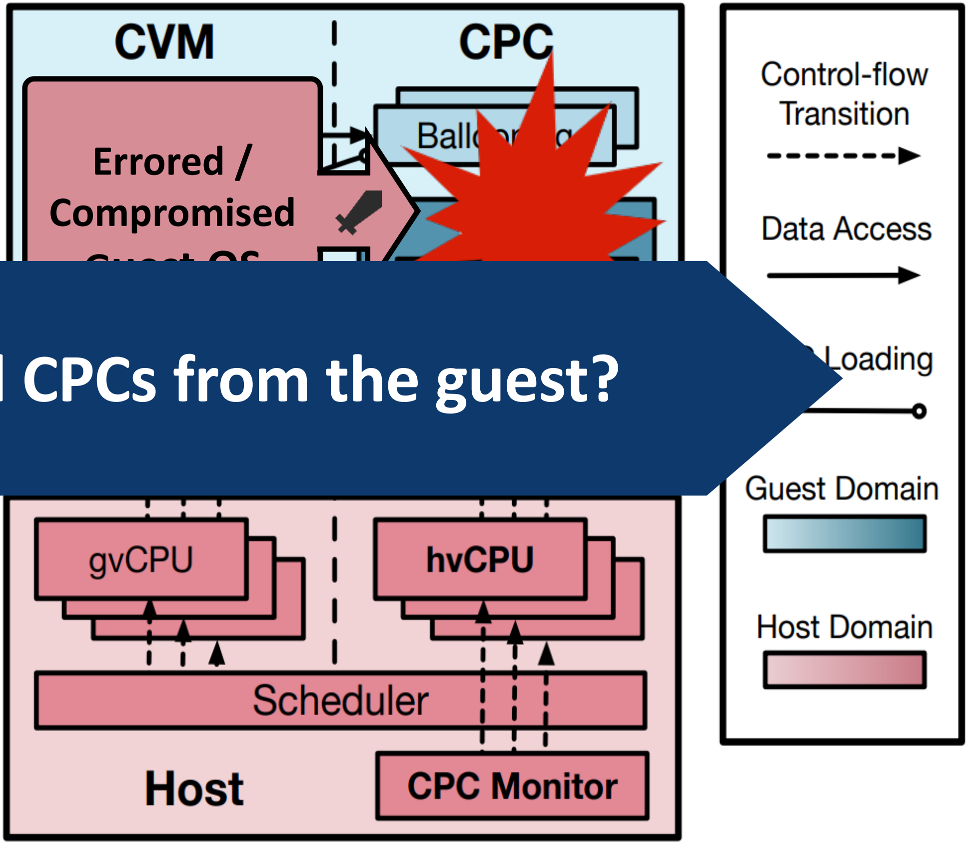


Tougher threat models than CVMs

- Part of maintenance modules should work correctly even when the guest OS errors
  - E.g., disaster recovery, data backup
    e

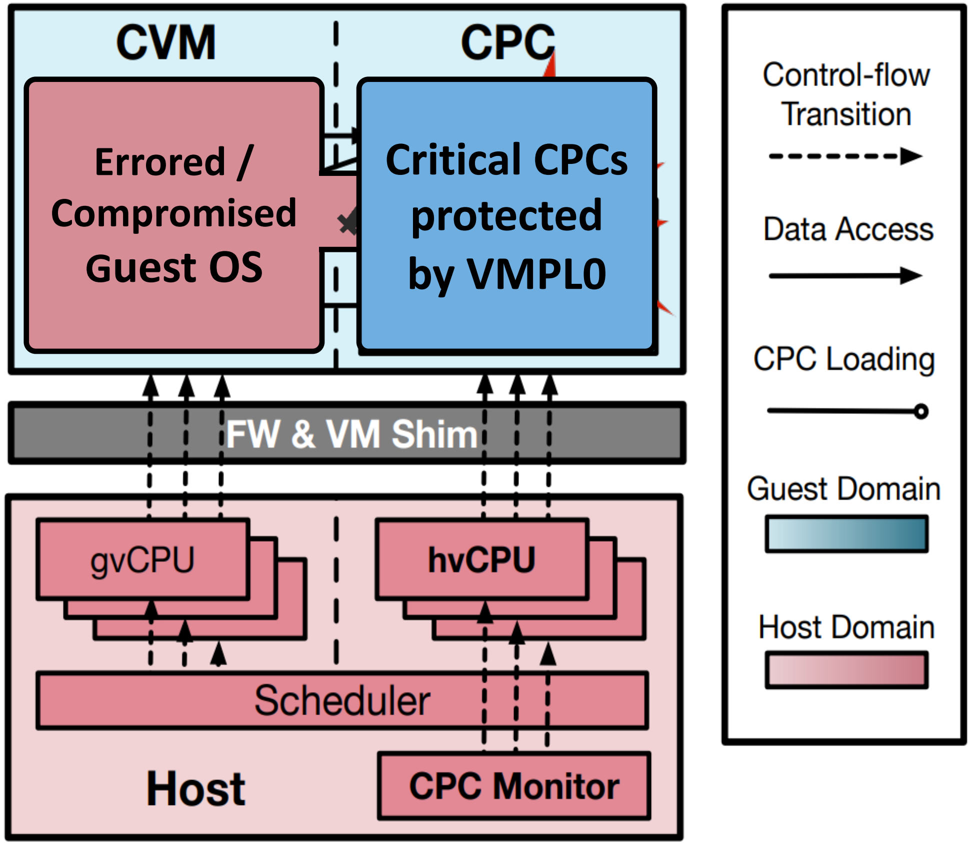
How to isolate critical CPCs from the guest?

- Huge T
  - Vulnerable, crash-prone, insecure
- Current CPC cannot survive tampering with a faulty guest OS
  - Resulting in damage to the CPCs, important data cannot be salvaged



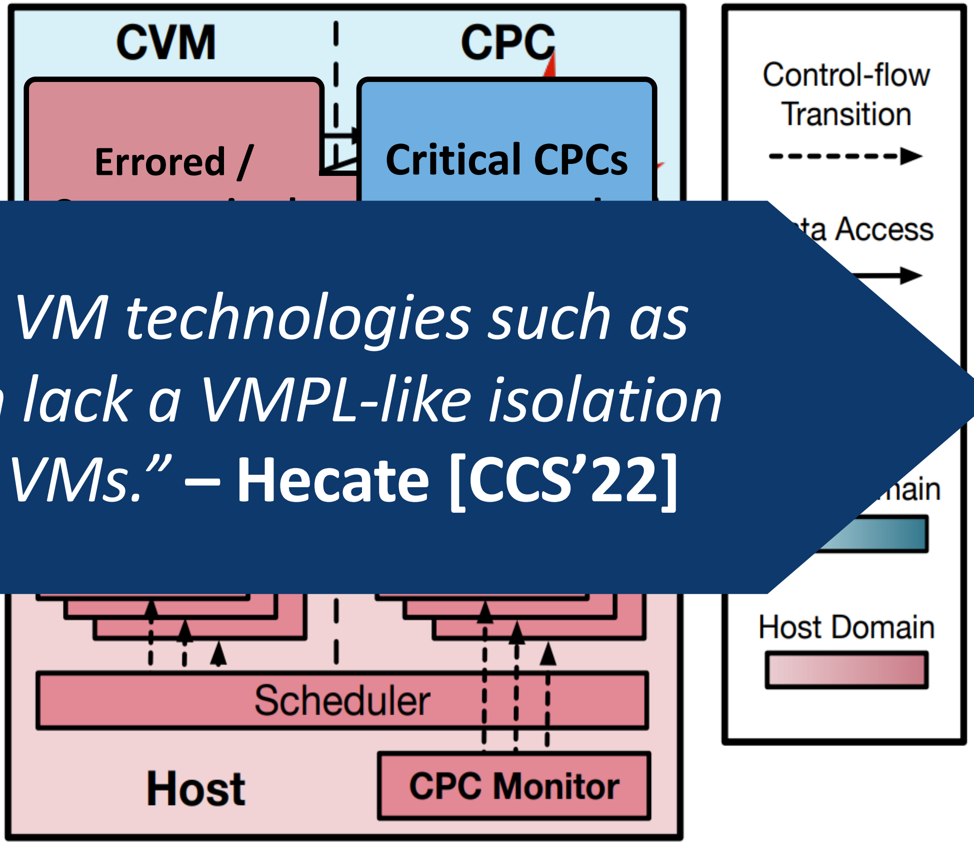
Virtual Machine Privilege Level

- AMD VMPL provides intra-CVM isolation
  - Guest OS on VMPL1-3 (Low privilege)
  - Critical CPCs on VMPL0 (High privilege)
  - Guest OS cannot access the memory and states of VMPL0
- Microsoft Hecate[CCS22] use VMPL to protect security services
  - E.g., firewall



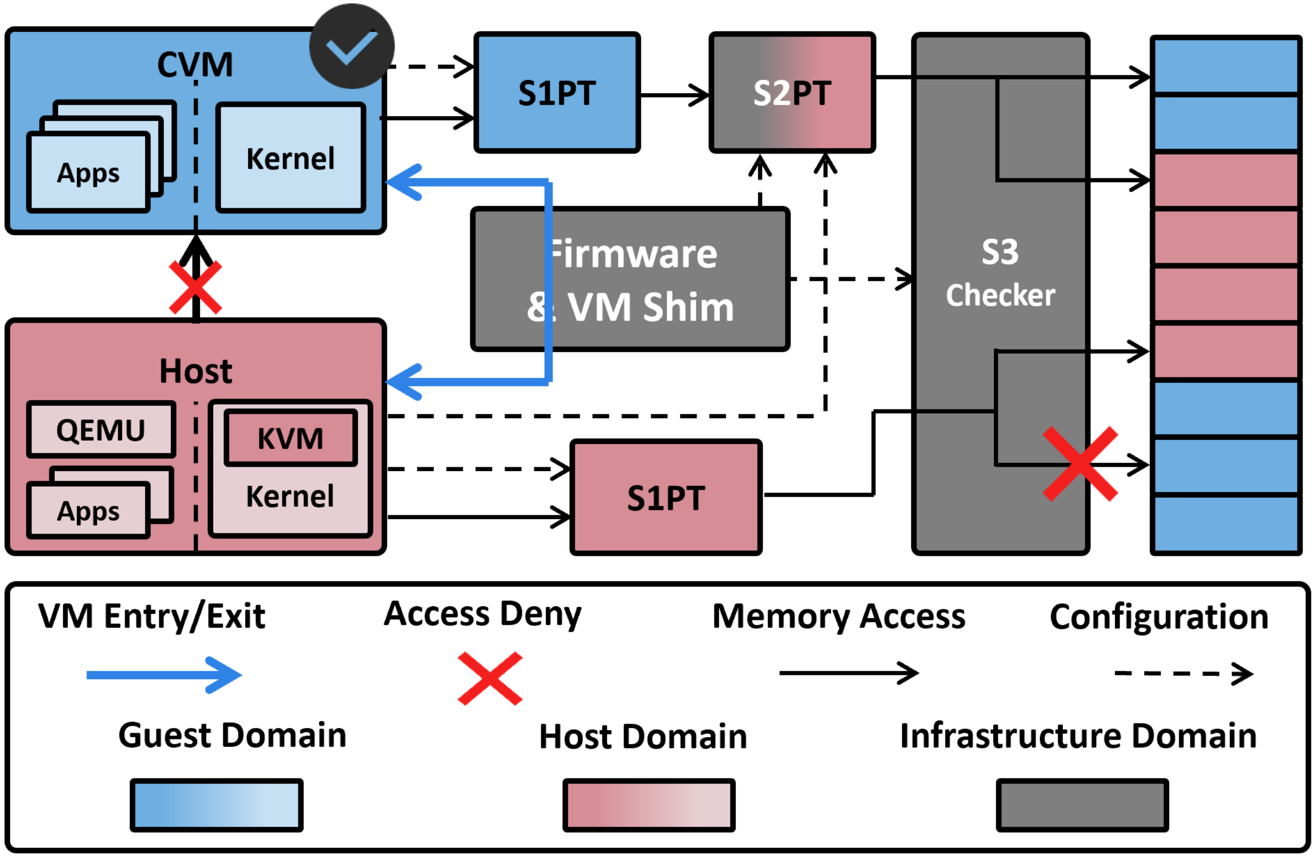
Virtual Machine Privilege Level

- AMD VMPL provides intra-CVM isolation
  - Guest OS on VMPL1-3 (Low privilege)
  - Gu
- to protect security services
  - E.g., firewall



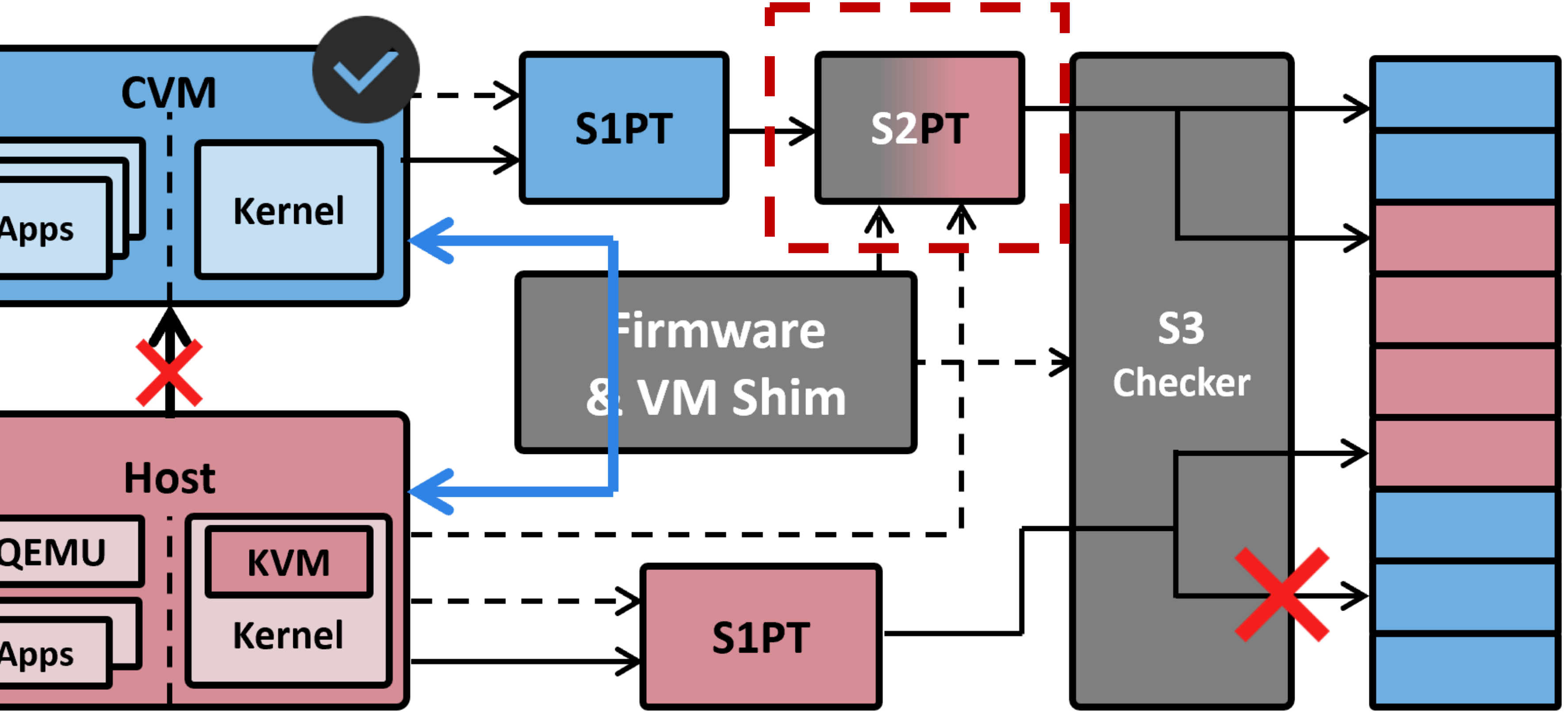
"Other new confidential VM technologies such as Intel TDX and ARM Realm lack a VMPL-like isolation inside their confidential VMs." **– Hecate [CCS'22]**

Only for AMD?



CVM
Apps
Kernel
S1PT
S2PT
Firmware
& VM Shim
S3
Checker
Host
QEMU
KVM
Apps
Kernel
S1PT
VM Entry/Exit
Access Deny
Memory Access
Configuration
Guest Domain
Host Domain
Infrastructure Domain

A Little Hope...

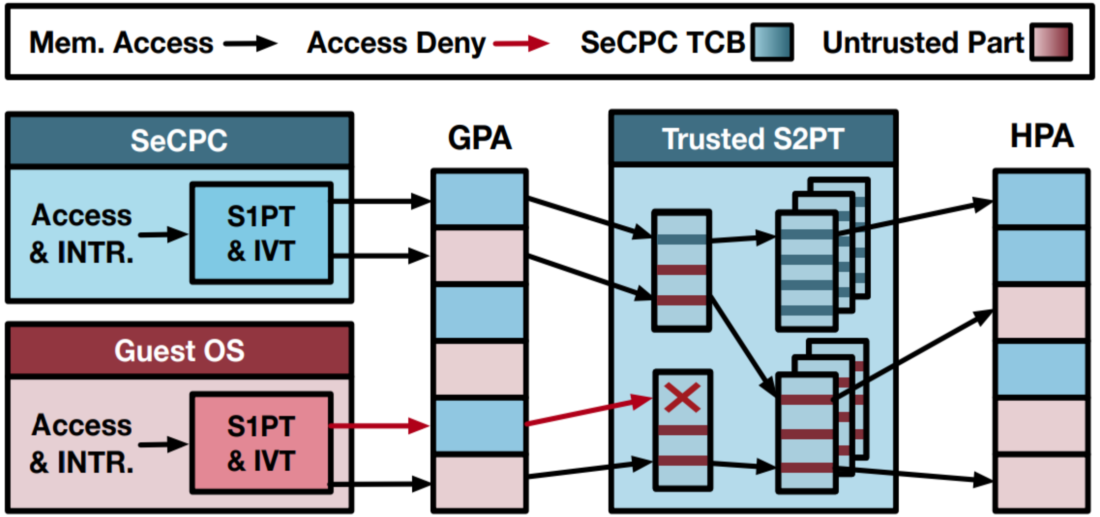


A Little Hope…

- AMD **has VMPL**
  - But the **S2PTs are controlled by the untrusted host**
- Intel TDX、ARM CCA、RISC-V CoVE **has no VMPL**
  - But the **S2PTs of CVM private memory are controlled by trusted components**

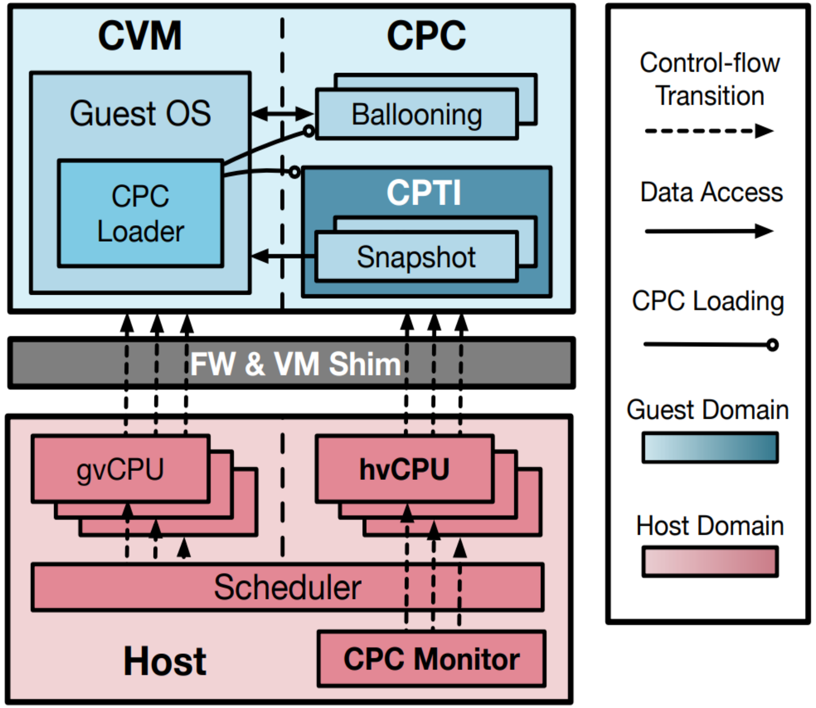
Confidential Page Table Isolation

- CPCs with CPTI → **SeCPCs (Secure CPC)**
- Isolated S2PTs only for the hvCPUs with SeCPCs are created by the trusted firmware
  - Mapping **extra trusted memory** for the SeCPC
  - S2PTs of gvCPUs have no such mappings



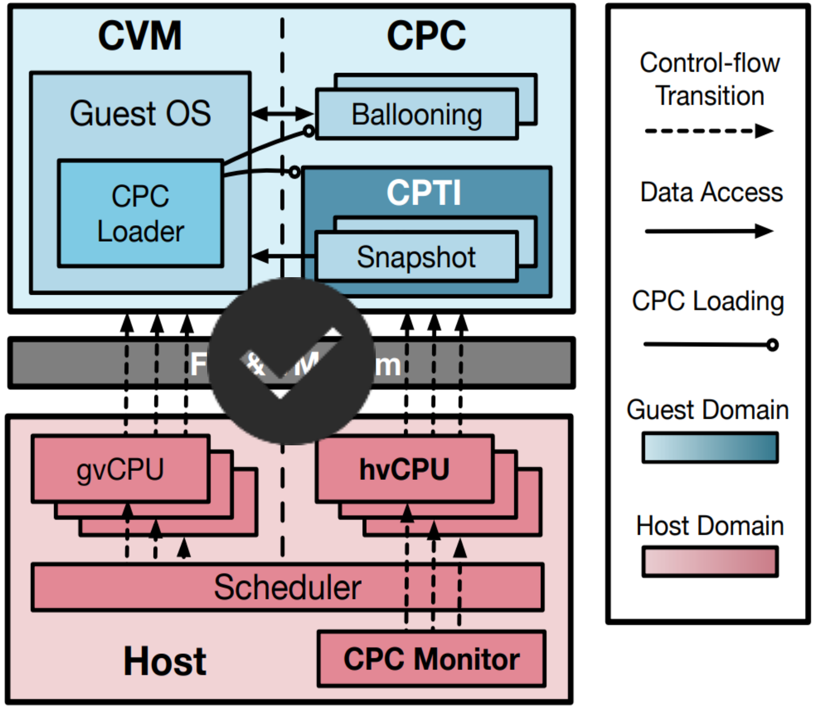
- SeCPC will further build its **trusted S1PT and IVT** in the trusted memory
  - On-demand TLB flush by FW & hvCPU register isolation

Security Analysis (on ARM CCA)



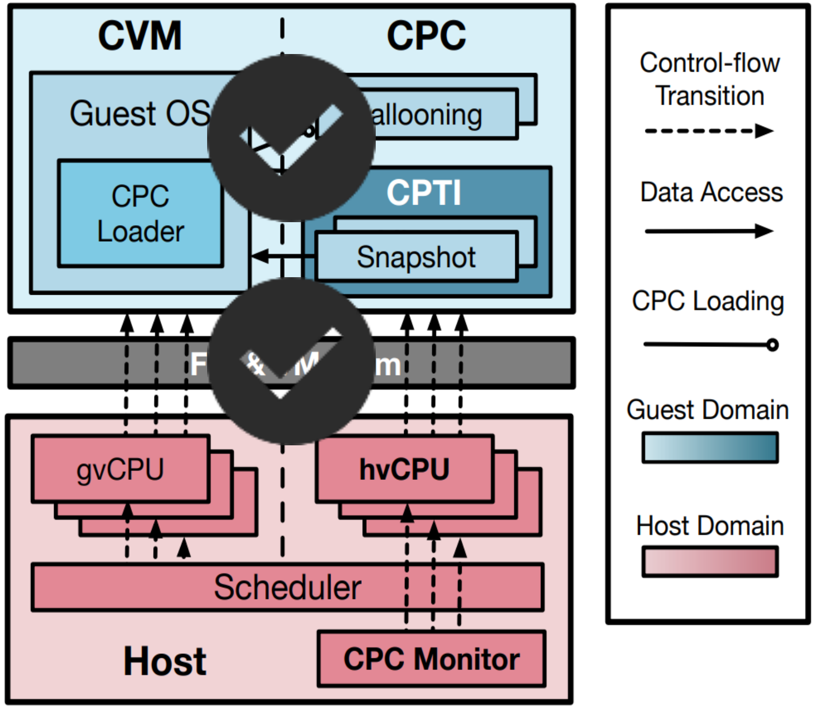
Security Analysis (on ARM CCA)

- Compacting firmware
  - Two simple maintenance modules (snapshot and security logging) will cause **7.23x** more modifications on the firmware



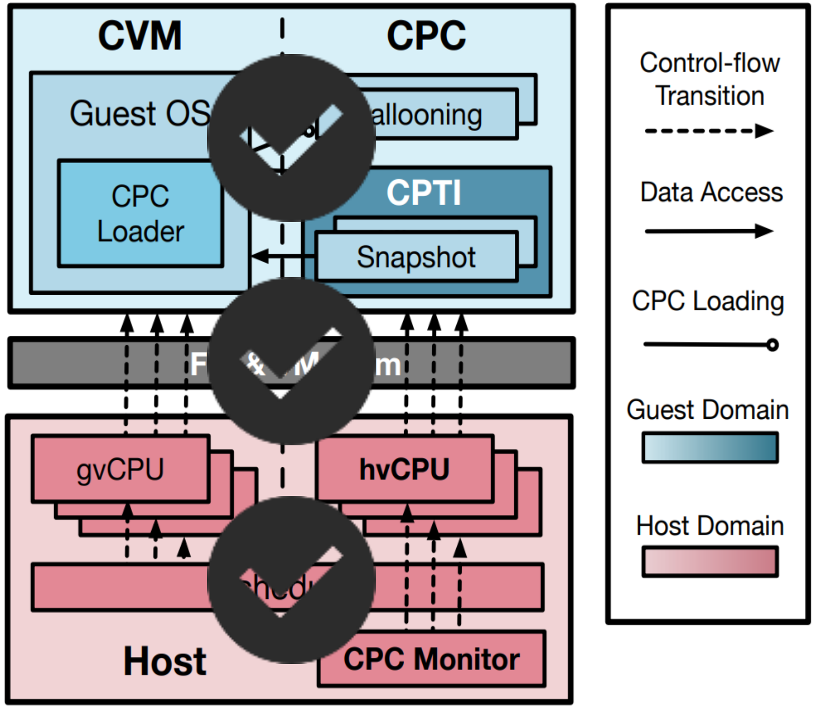
Security Analysis (on ARM CCA)

- Compacting firmware
  - Two simple maintenance modules (snapshot and security logging) will cause **7.23x** more modifications on the firmware
- Guest security:
  - CPC code size is small compared to Linux
  - CPC can be timely patched
  - CPTI can also prevent an errored CPC
  - Only equip the CPCs that are really needed



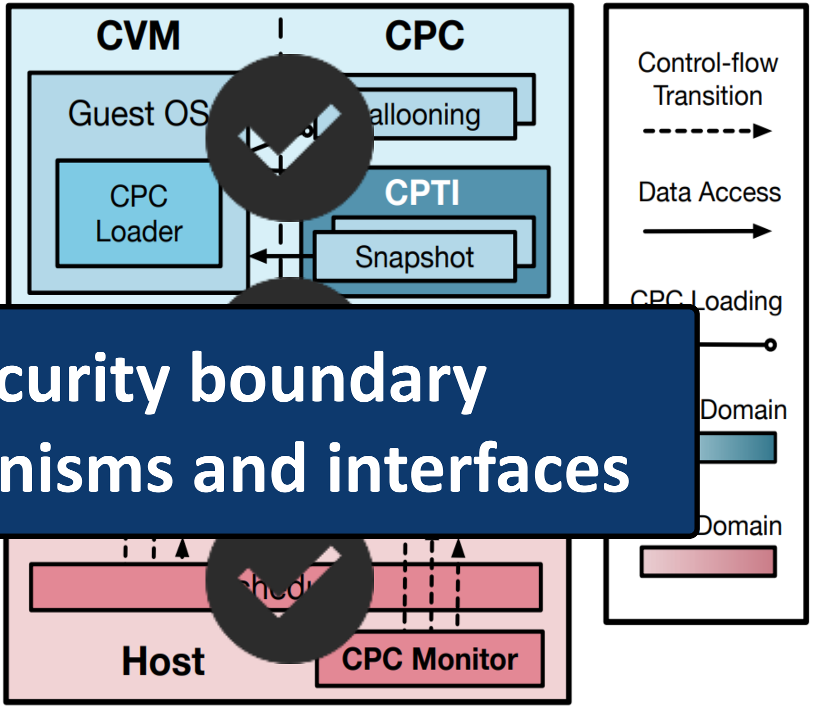
Security Analysis (on ARM CCA)

- Compacting firmware
  - Two simple maintenance modules (snapshot and security logging) will cause **7.23x** more modifications on the firmware
- Guest security:
  - CPC code size is small compared to Linux
  - CPC can be timely patched
  - CPTI can also prevent an errored CPC
  - Only equip the CPCs that are really needed
- Host security:
  - Few modifications in KVM, most of the modifications are in QEMU & KVMTOOL in the user space



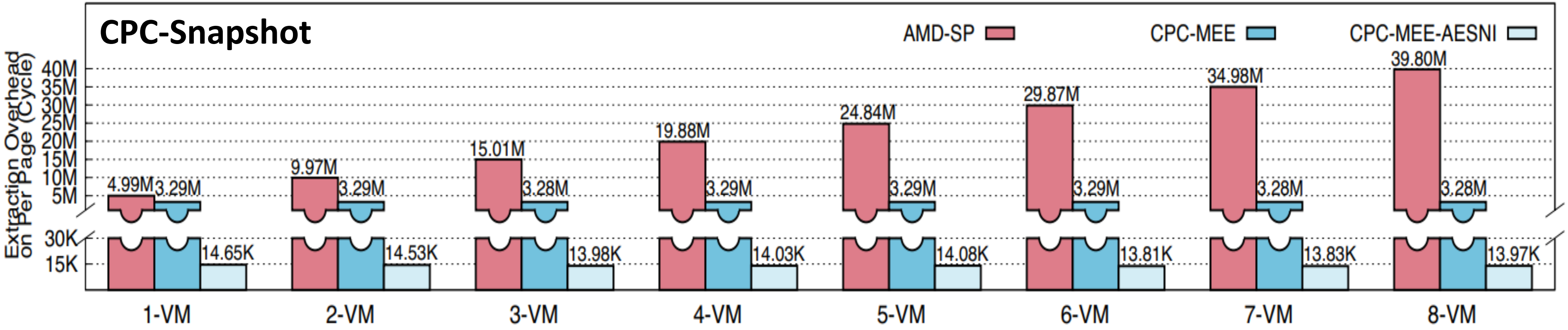
Security Analysis (on ARM CCA)

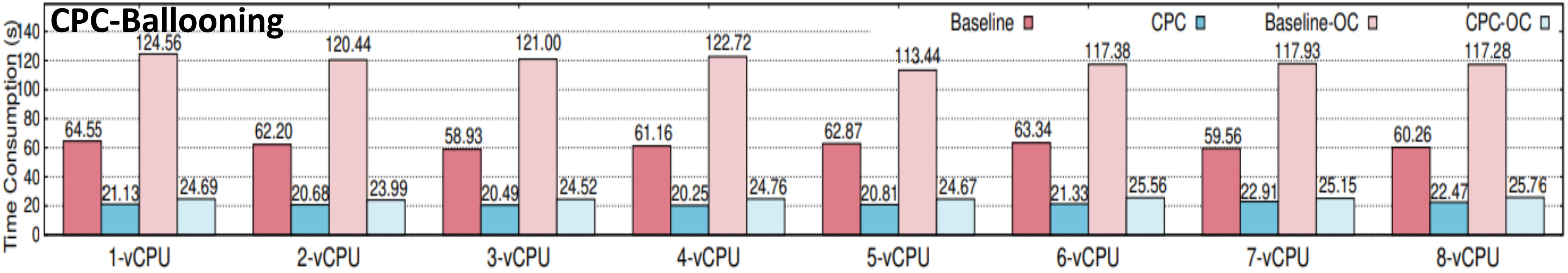
- Compacting firmware
  - Two simple maintenance modules (snapshot and security logging) will cause **7.23x** more modifications on the firmware
- Gu
  - Only equip the CPCs that are really needed
- Host security:
  - Few modifications in KVM, most of the modifications are in QEMU & KVMTOOL in the user space



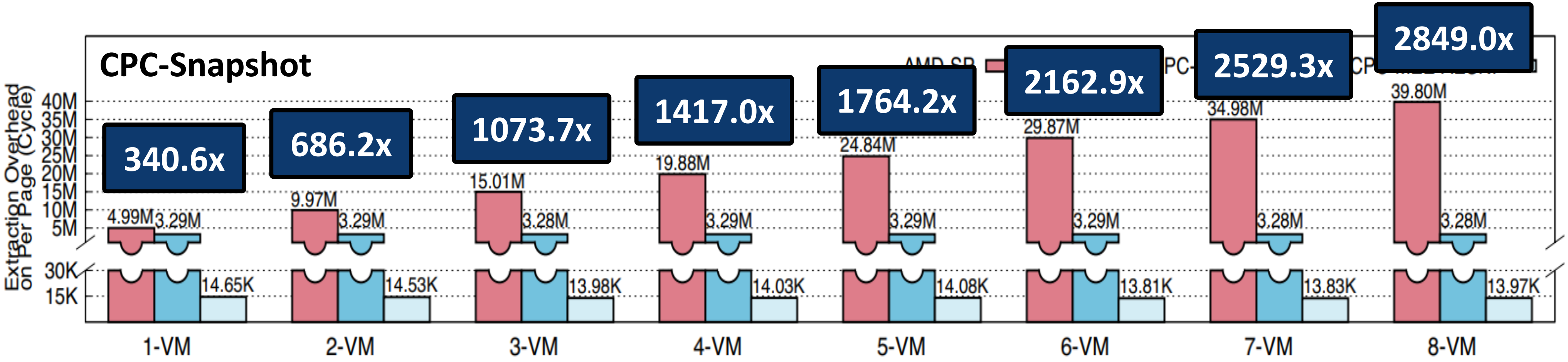
Maintain the clear security boundary
Reuse current mature mechanisms and interfaces

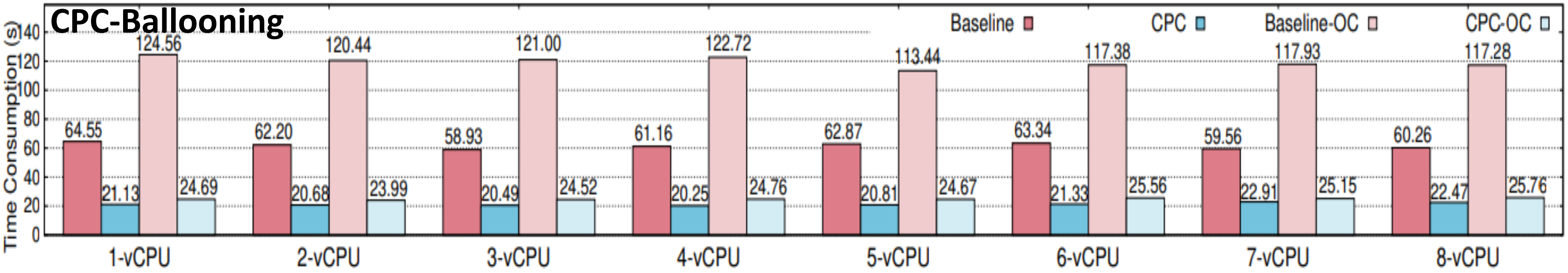
Performance Evaluation (on AMD SEV)



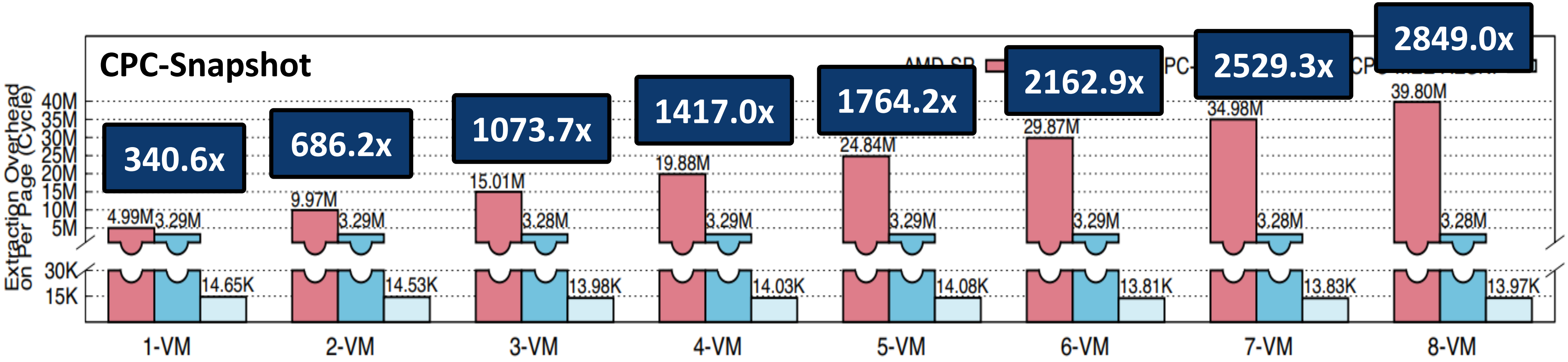


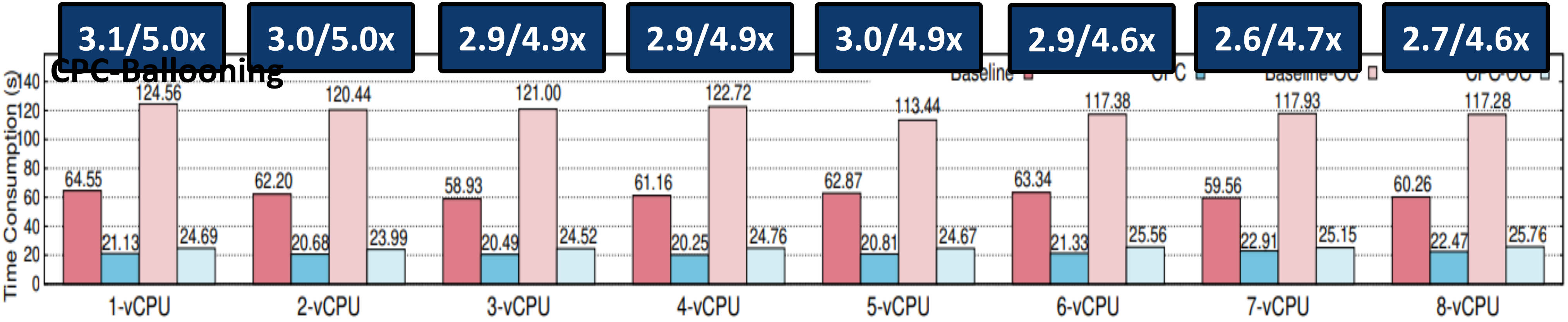
Performance Evaluation (on AMD SEV)





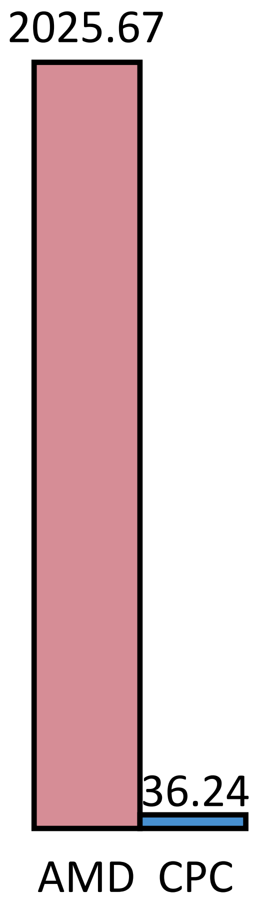
Performance Evaluation (on AMD SEV)





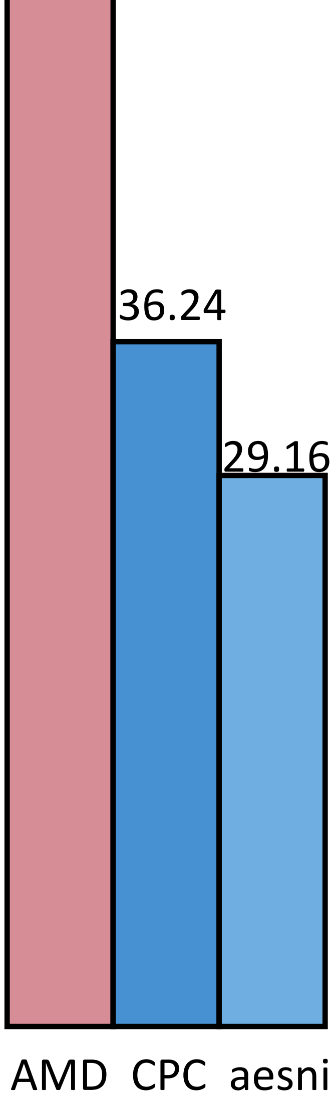
Migration Evaluation

- CPC-LiveMigration vs. AMD Solution:
  - AES-GCM in software (mbedtls), **55.90x faster**



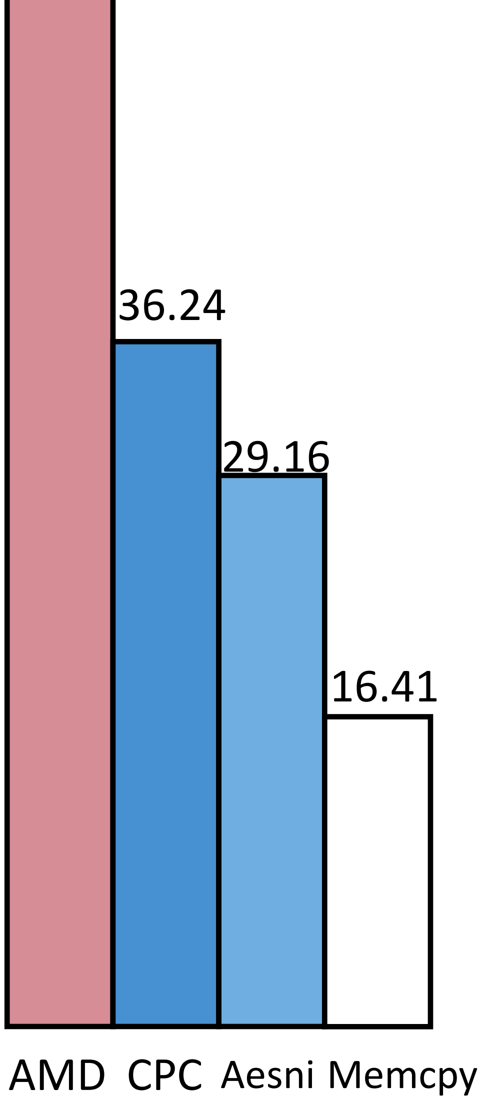
Migration Evaluation

- CPC-LiveMigration vs. AMD Solution:
  - AES-GCM in software (mbedtls), **55.90x faster**
  - With AESNI, **69.47x faster**



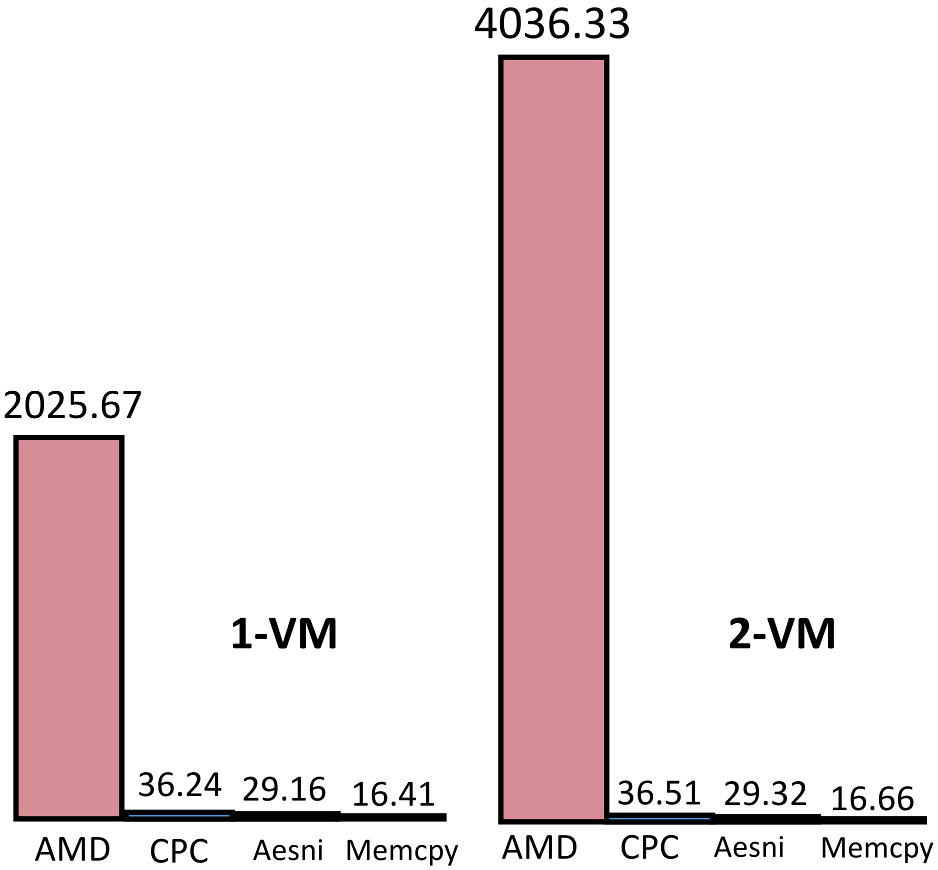
Migration Evaluation

- CPC-LiveMigration vs. AMD Solution:
  - AES-GCM in software (mbedtls), **55.90x faster**
  - With AESNI, **69.47x faster**
  - Upper bound of current CVM architecture?
    - AES-GCM → memcpy
    - 16.41s, **123.44x faster**



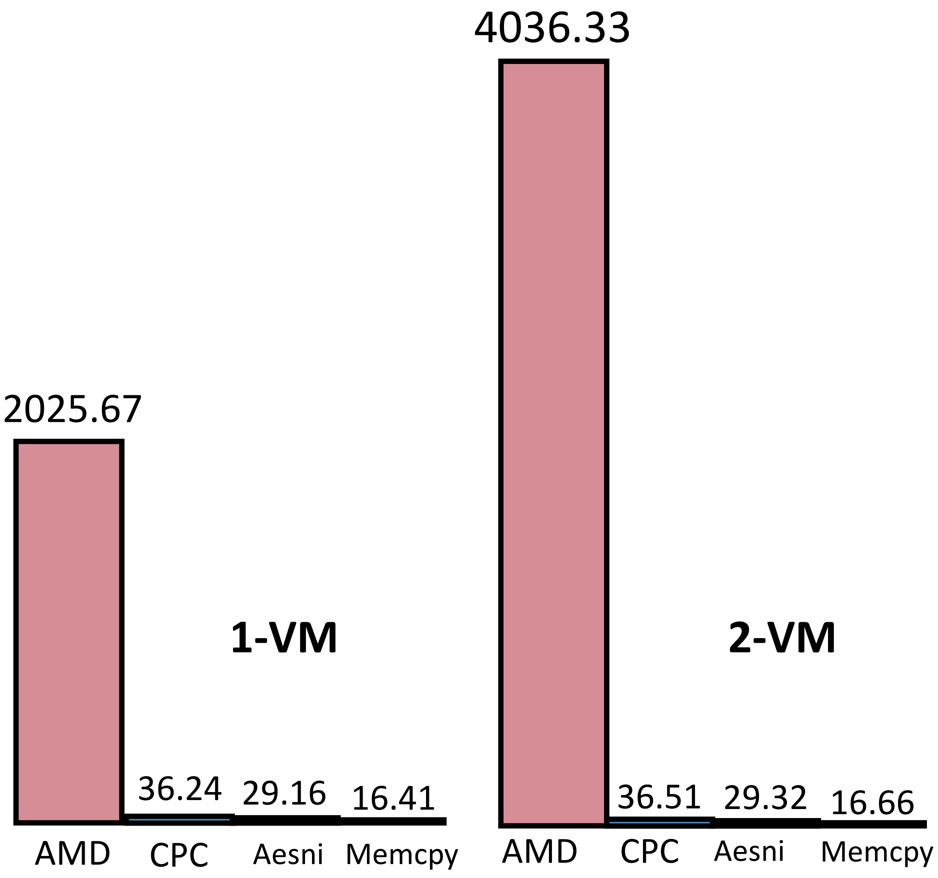
Migration Evaluation

- CPC-LiveMigration vs. AMD Solution:
  - AES-GCM in software (mbedtls), **55.90x faster**
  - With AESNI, **69.47x faster**
  - Upper bound of current CVM architecture?
    - AES-GCM → memcpy
    - 16.41s, **123.44x faster**
  - More instances, more improvements



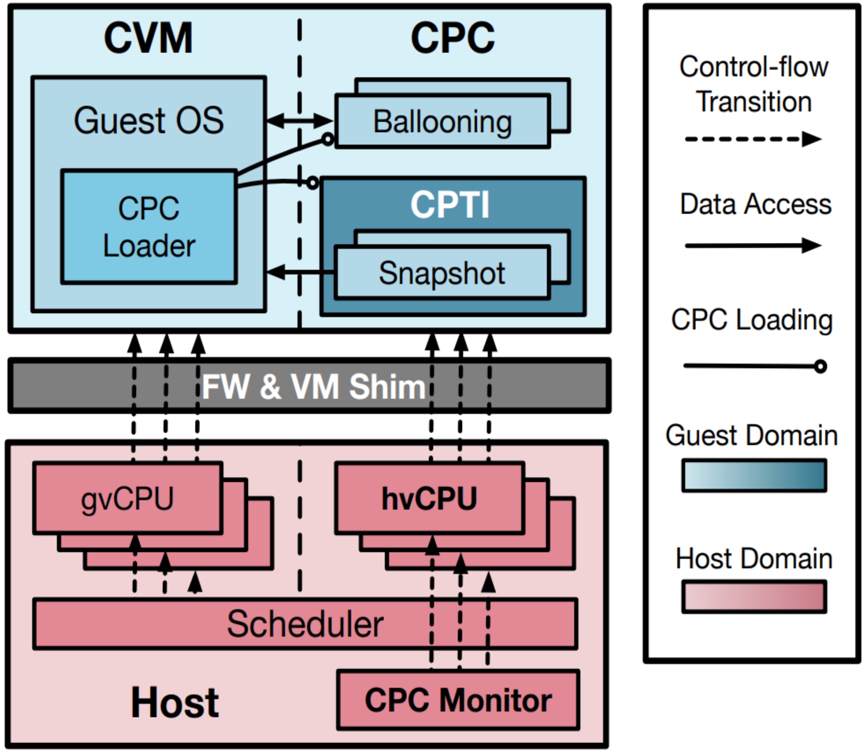
Migration Evaluation

- CPC-LiveMigration vs. AMD Solution:
  - AES-GCM in software (mbedtls), **55.90x faster**
  - With AESNI, **69.47x faster**
  - Upper bound of current CVM architecture?
    - AES-GCM → memcpy
    - 16.41s, **123.44x faster**
  - More instances, more improvements
- Further acceleration:
  - Multi-threading (multifd)
  - Post-copy
    - Current AMD-SP cannot support, but CPCs can



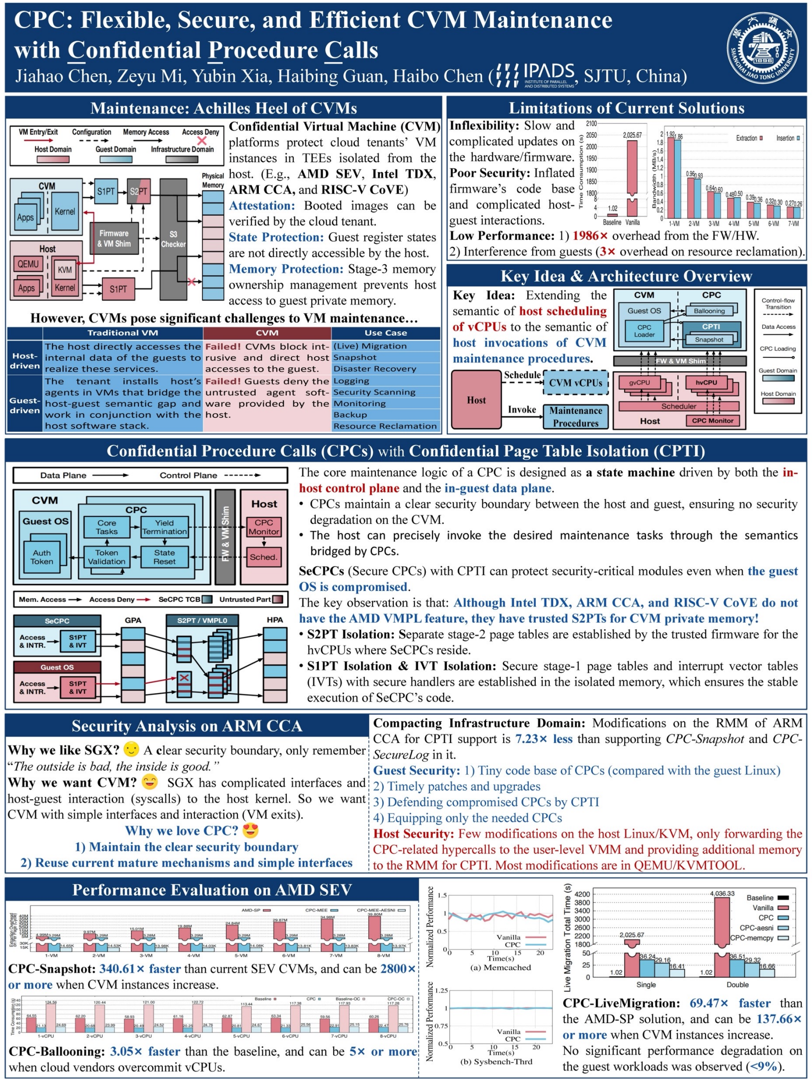
Conclusion

- Confidential Procedure Calls
  - Extend the semantics of **vCPU scheduling** into the semantics of **host invocations of maintenance procedures**
- A more **flexible**, **secure**, and **efficient** CVM maintenance solution
  - Enable customized maintenance modules defined by the cloud tenants and vendors
  - Maintain clear security boundary and reuse mature mechanisms
  - Achieve significant performance improvements
- Compatible with all current CVM platforms



Thanks
Q&A

chenjiahaosys@gmail.com



Thanks

Q&A

aosys@gmail.com

IPADS
INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEMS

Get the Poster

WeChat

Thanks
Q&A

chenjiahaosys@gmail.com

The upcoming lunch
is on your own~

SHANGHAI JIAO TONG UNIVERSITY 1896

IPADS
INSTITUTE OF PARALLEL
AND DISTRIBUTED SYSTEMS

Backup

INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEMS

Different Meaning for Different CVMs

Cloud Provider & Tenants

Different CVMs, One Answer

CPC: **C**onfidential **P**rocedure **C**alls

Cloud Provider & Tenants

Optimization

Optimization 1: Following the philosophy of separating the control plane from the data plane.

Optimization 2: Reusing generic operators in multiple scenarios.

Optimization 3: Open sourcing for public validation.

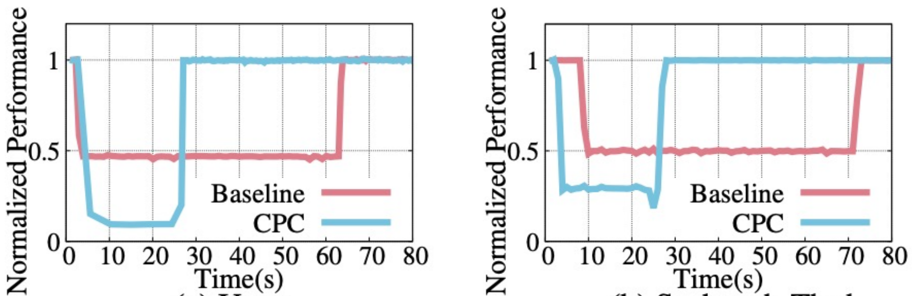
Optimization 4: Distinction between CPC and SeCPC scenarios.

Table 2: Description of generic maintenance operators.

Name	Description
Memory Encryption Extraction (**MEE**)	Encrypt and extract the private data from the target GPA to the host domain.
State Encryption Extraction (**SEE**)	Encrypt and extract the private states from the target vCPU to the host domain.
Memory Decryption Insertion (**MDI**)	Insert and decrypt the private data to the target GPA in CVM.
State Decryption Insertion (**SDI**)	Insert and decrypt the private states to the target vCPU in CVM.

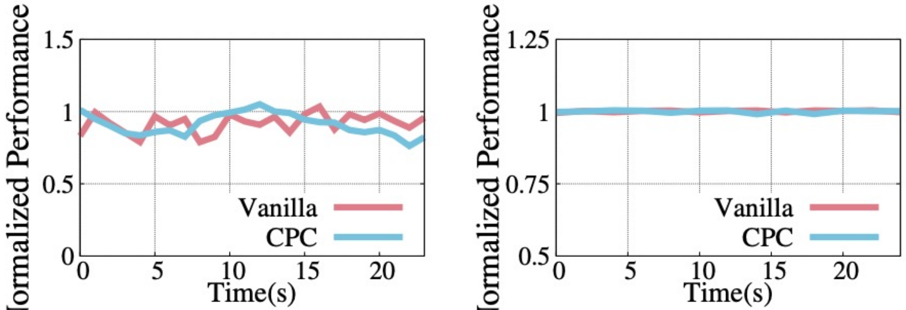
Resource Isolation

CPCs offer isolated CPU resources for maintenance modules, and SeCPCs can additionally provide memory isolation. However, in certain scenarios, maintenance modules may need to leverage the internal data structure and semantics of the guest OS. Consequently, they cannot be completely isolated from the guest workloads, as shown in Memory Reclamation test on the left. In the migration test, this isolation is complete.



(a) Untar (b) Sysbench-Thrd

Memory Reclamation

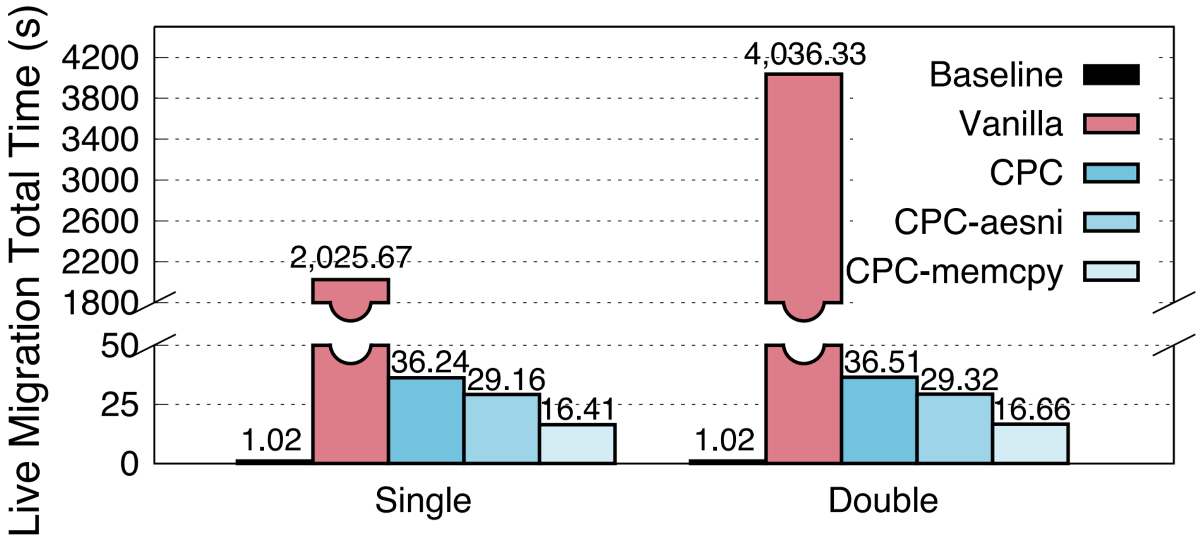


(a) Memcached (b) Sysbench-Thrd

Live Migration

Performance Evaluation

- CPC-LiveMigration vs. AMD solution:
  - Without AESNI? **55.90x faster**
  - With AESNI? **69.47x faster**, still gap from traditional VMs
    - Overhead mainly from GCM, not AES
  - 2-CVM? **Double** the improvement
    - AMD-SP is shared out



  - Upper Bound if CVM: memcpy can achieve **123.44 faster** even with 1-CVM migration
    - Assume that we can develop a hardware that makes AES & GCM as fast as a simple memcpy
  - Future work: Multi-threading (multifd)，Async/Pipeline，Post-copy (AMD-SP cannot support this, but CPC can)

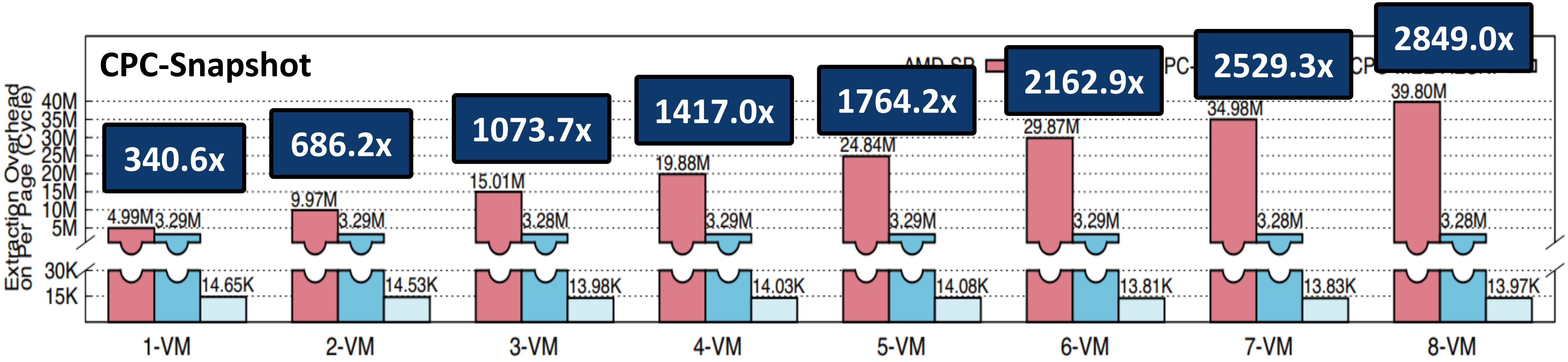
Confidential Abort Protocol

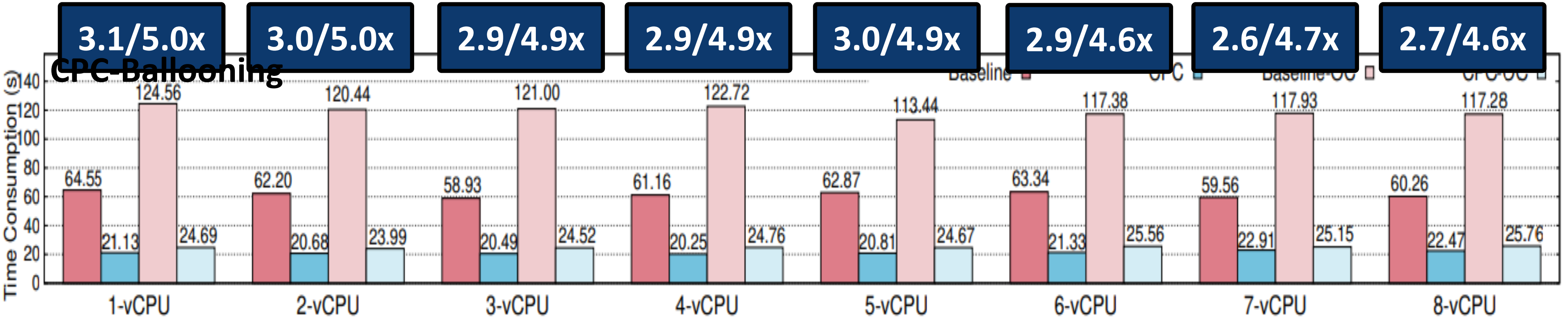
The basic idea is that dishonest tenants only hurt themselves.

For a CPC-Reclamation, the host only needs to set a throughput threshold based on the economic value of the reclaimed resources. When the CPC cannot provide a sufficient amount of reclaimed resources, the host assumes that the free resources in the guest are depleted and stops CPCReclamation. A dishonest guest cannot excessively divert resources from the hvCPU to avoid reclaiming below the threshold. On the other hand, if it deceptively commits unrecoverable resources to the host to boost throughput, it will error out due to those resources being taken without any damage to the host.

In the case of CPC-Migration, the host can set a migration time limit. Specifically, since the migration time is proportional to the size of the guest memory, the host can accurately estimate the reasonable CPU time that CPC-Migration should occupy. When the time limit expires, the host just deschedules the CPC-Migration. A dishonest guest that over-appropriates hvCPU resources will cause the migration to not complete, resulting in errors in its destination instance.

Performance Evaluation (on AMD SEV)





END

SHANGHAI JIAO TONG UNIVERSITY

IPADS
INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEMS