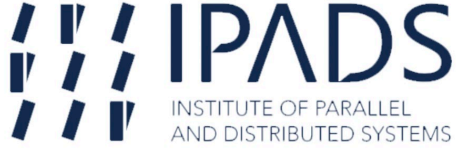
第26届中国计算机系统研讨会
ChinaSys’24 Spring

CPC: 用机密过程调用来实现灵活、安全和高性能的机密虚拟机运维

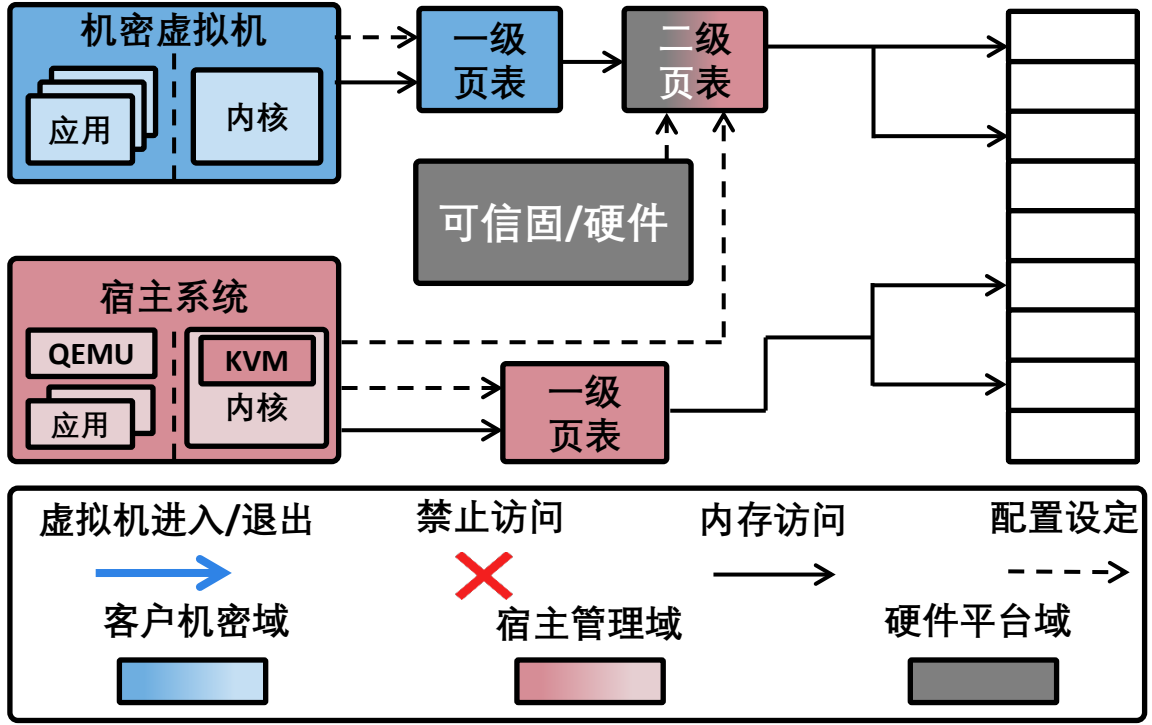
陈家浩，糜泽羽，夏虞斌，管海兵，陈海波

上海交通大学 并行与分布式系统研究所



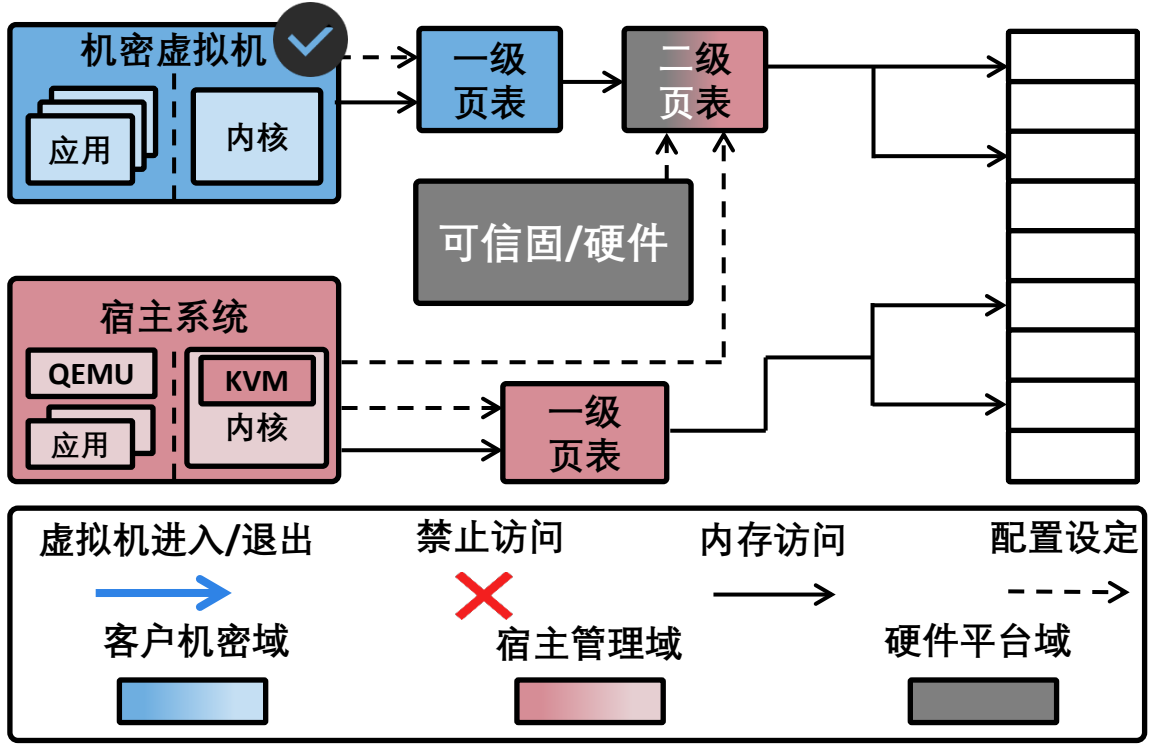
机密虚拟机（CVM）保护云租户隐私

- CVM--把VM放进TEE中



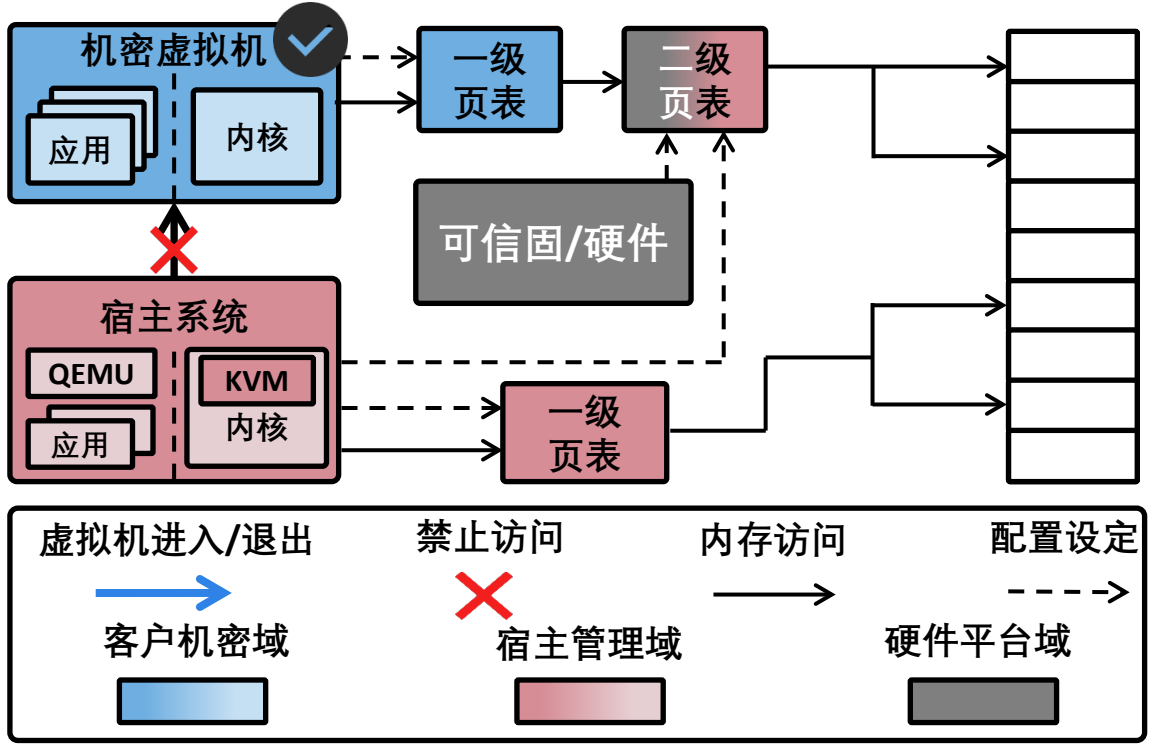
机密虚拟机（CVM）保护云租户隐私

- CVM--把VM放进TEE中
  - VM的启动镜像可被验证



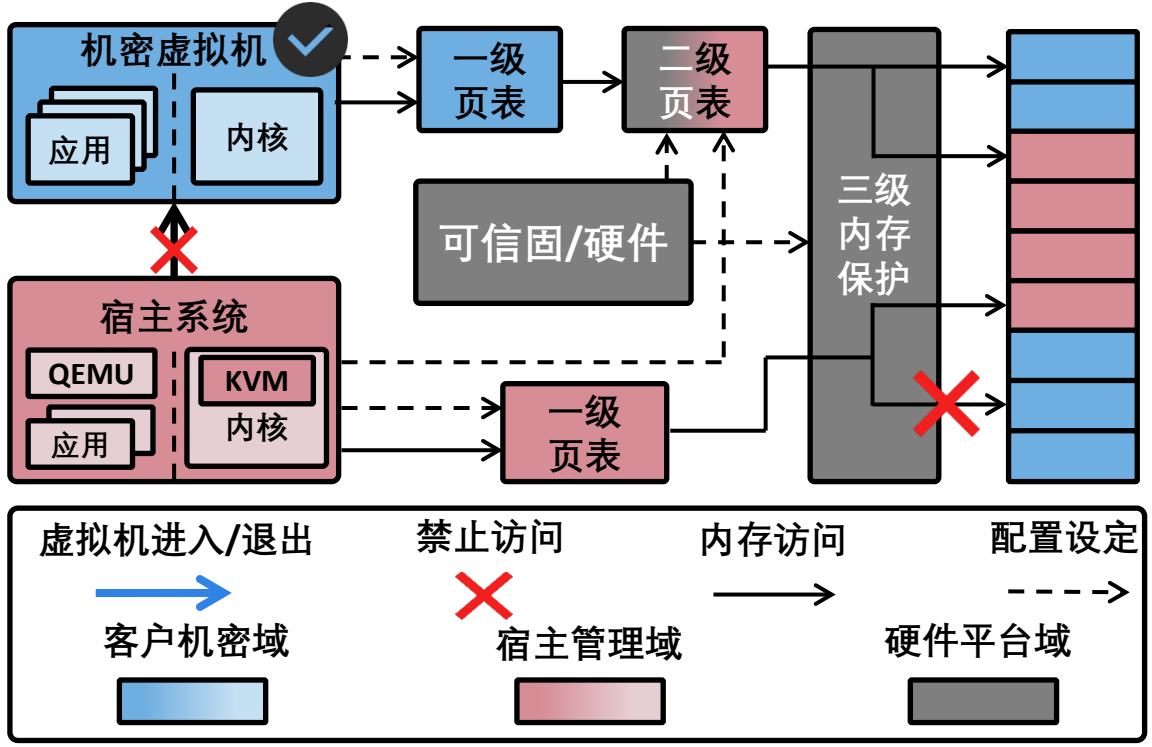
机密虚拟机（CVM）保护云租户隐私

- CVM--把VM放进TEE中
  - VM的启动镜像可被验证
  - 客户寄存器状态不能被宿主机直接读取



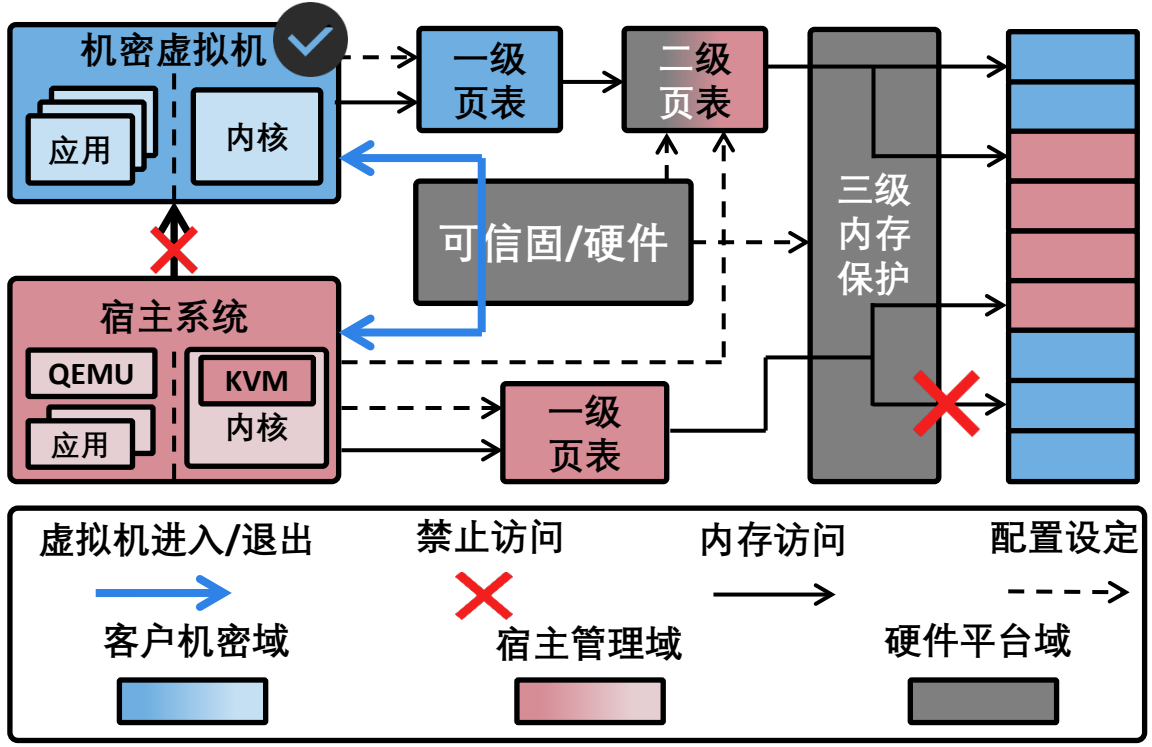
机密虚拟机（CVM）保护云租户隐私

- CVM--把VM放进TEE中
  - VM的启动镜像可被验证
  - 客户寄存器状态不能被宿主机直接读取
  - 三级内存保护和加密使数据不能被宿主机访问



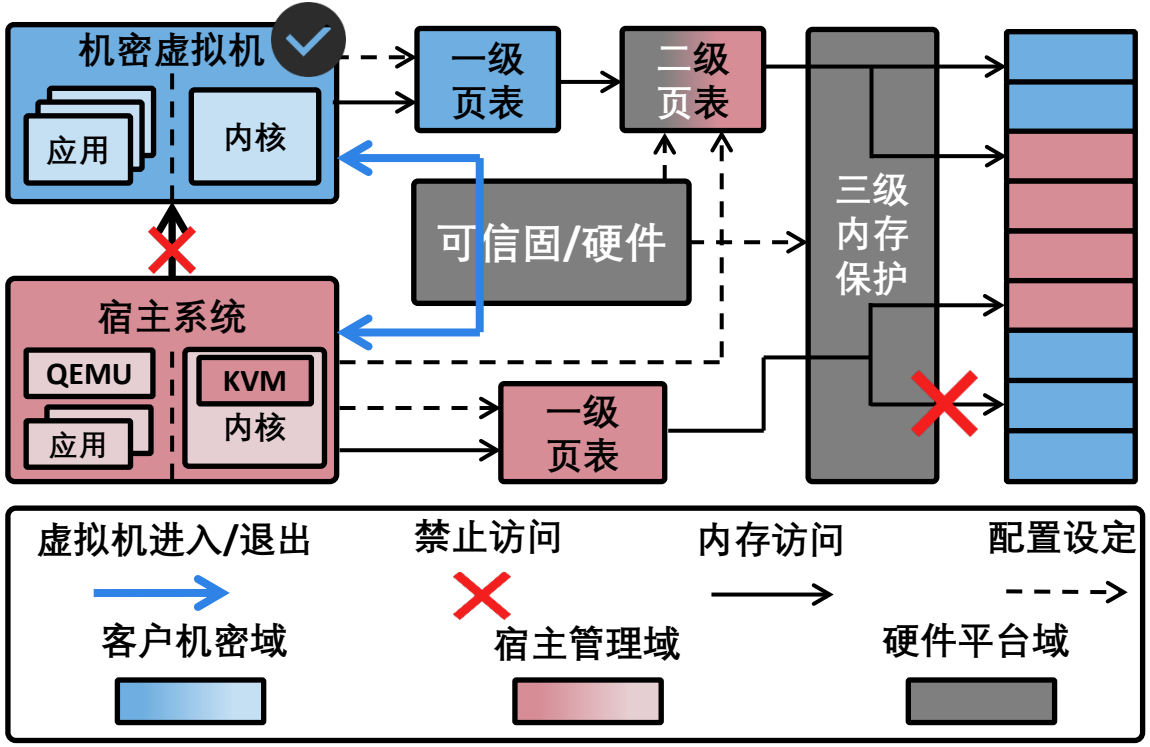
机密虚拟机（CVM）保护云租户隐私

- CVM--把VM放进TEE中
  - VM的启动镜像可被验证
  - 客户寄存器状态不能被宿主机直接读取
  - 三级内存保护和加密使数据不能被宿主机访问
  - CVM的下陷会被可信固/硬件过滤



机密虚拟机（CVM）保护云租户隐私

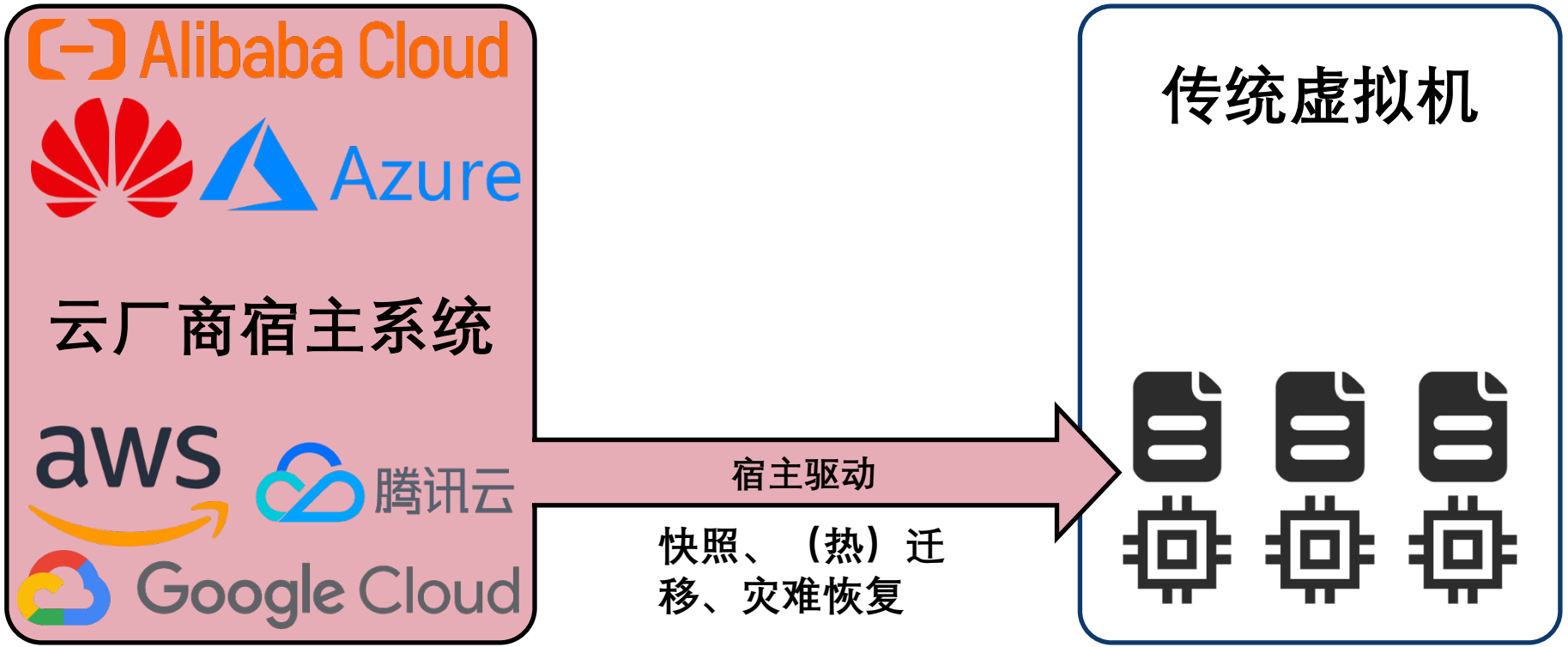
- CVM--把VM放进TEE中
  - VM的启动镜像可被验证
  - 客户寄存器状态不能被宿主机直接读取
  - 三级内存保护和加密使数据不能被宿主机访问
  - CVM的下陷会被可信固/硬件过滤



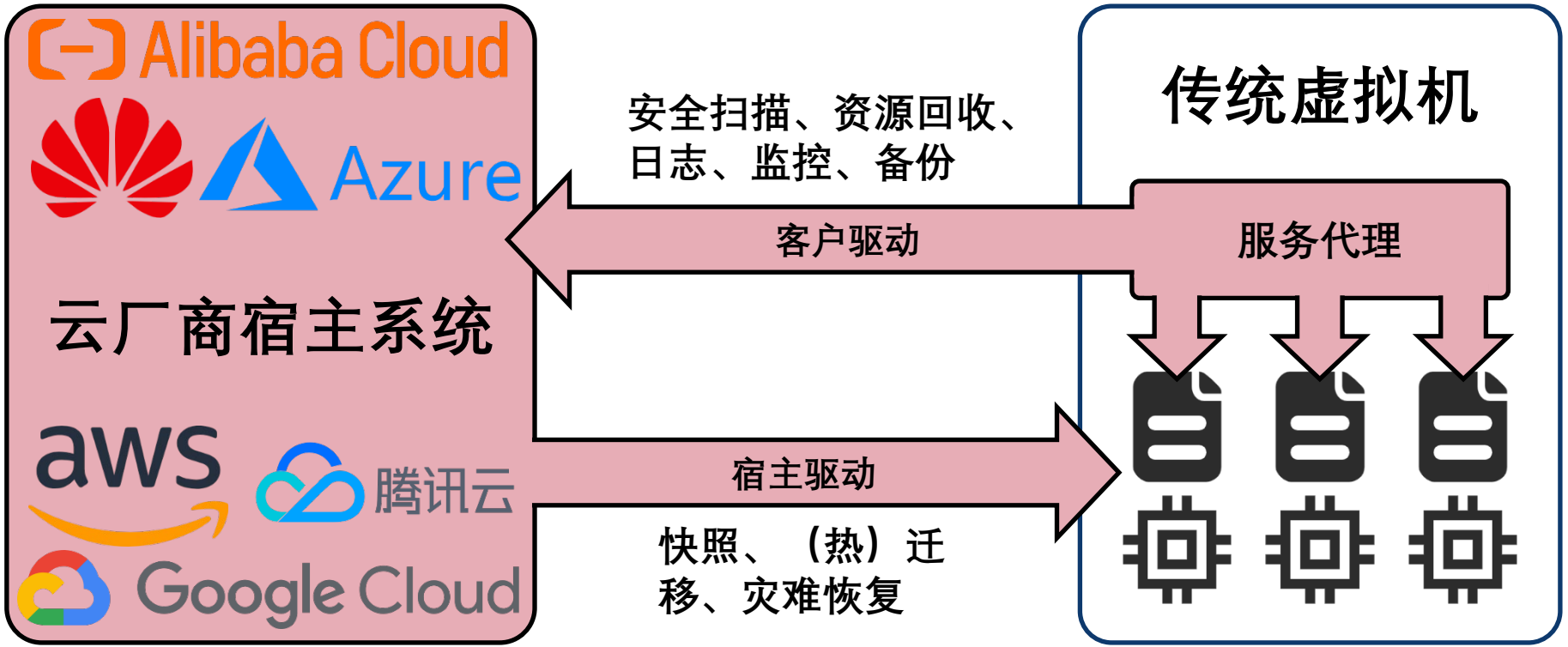
- 主流的CVM平台：**AMD SEV, Intel TDX, ARM CCA 和 RISC-V CoVE**

但运维是个大问题...

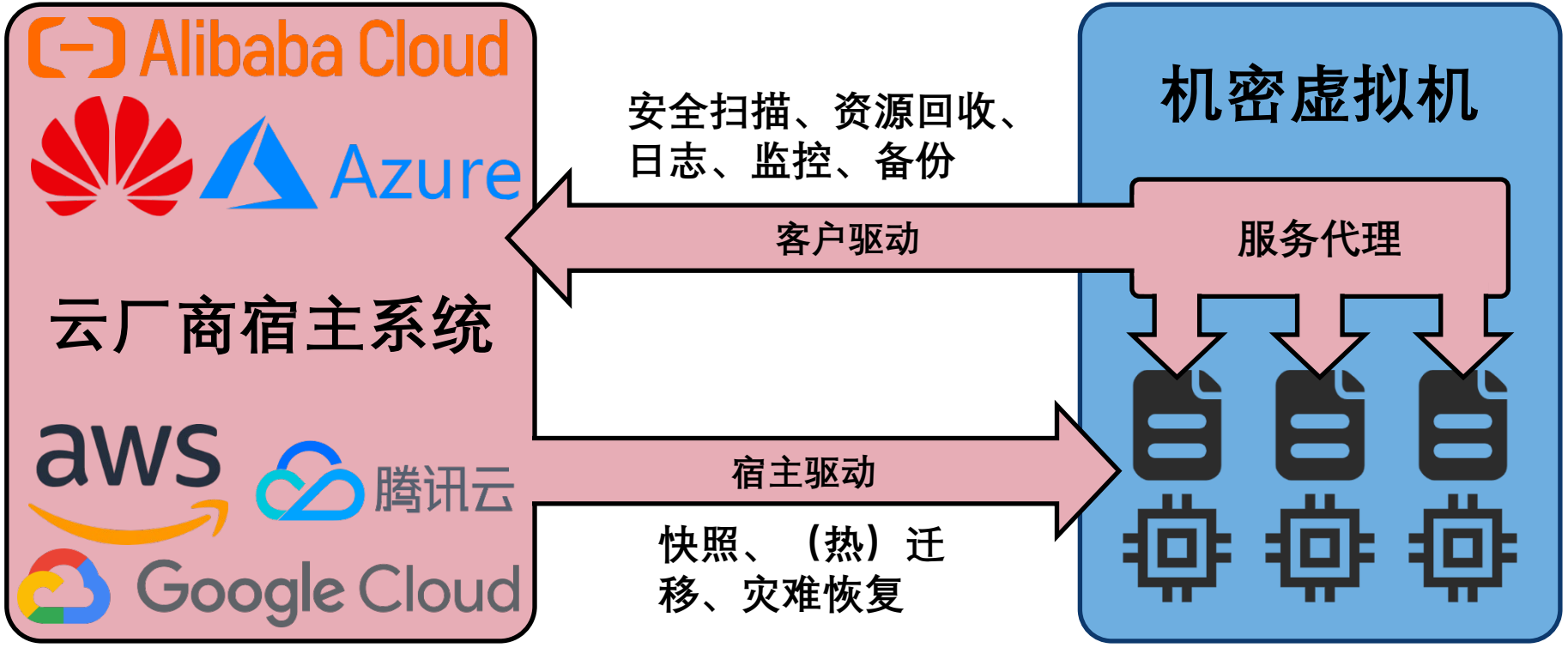
但运维是个大问题...


Alibaba Cloud
Azure
云厂商宿主系统
aws
腾讯云
Google Cloud
宿主驱动
快照、（热）迁移、灾难恢复
传统虚拟机

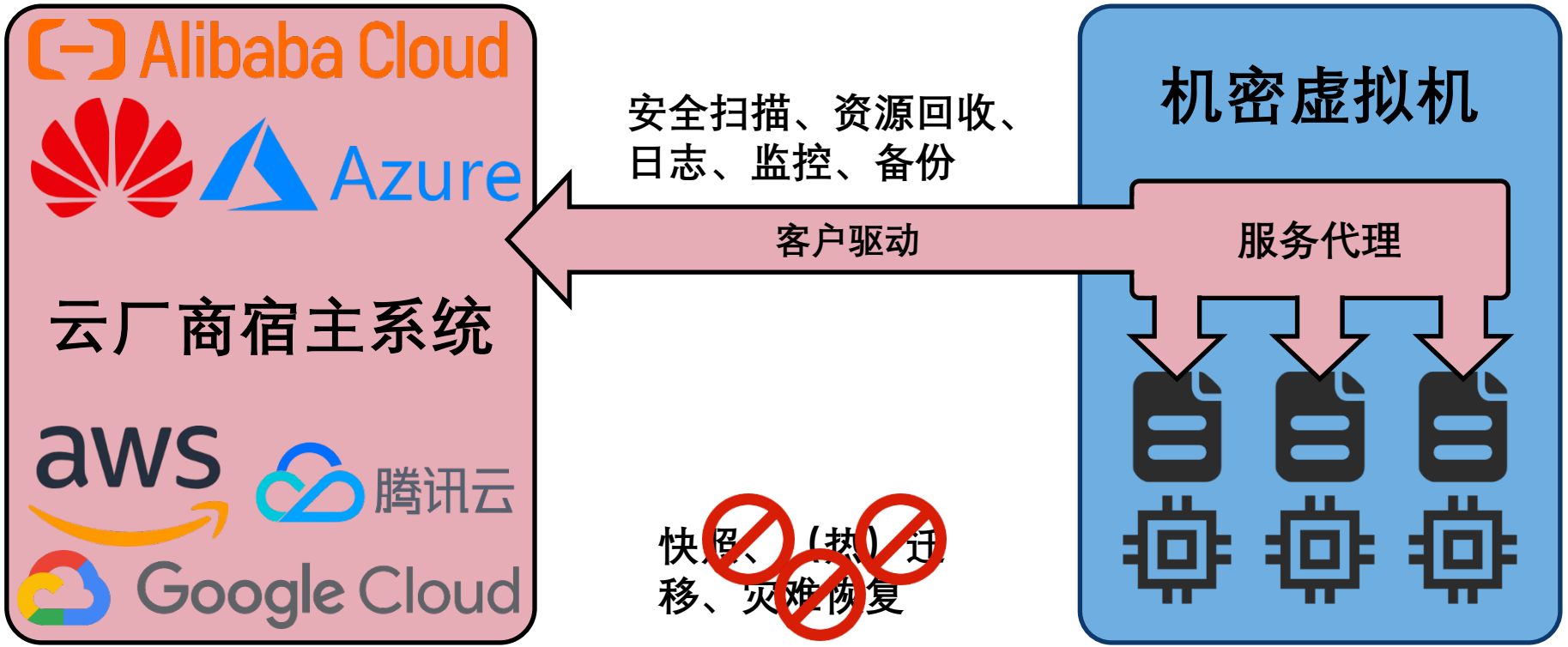
但运维是个大问题...


Alibaba Cloud
Azure
云厂商宿主系统
aws
腾讯云
Google Cloud
安全扫描、资源回收、
日志、监控、备份
客户驱动
宿主驱动
快照、（热）迁
移、灾难恢复
传统虚拟机
服务代理

但运维是个大问题...


Alibaba Cloud
Azure
云厂商宿主系统
aws
腾讯云
Google Cloud
安全扫描、资源回收、
日志、监控、备份
客户驱动
宿主驱动
快照、（热）迁
移、灾难恢复
机密虚拟机
服务代理

但运维是个大问题...


Alibaba Cloud
Azure
云厂商宿主系统
aws
腾讯云
Google Cloud
安全扫描、资源回收、
日志、监控、备份
客户驱动
机密虚拟机
服务代理
移、灾难恢复

但运维是个大问题...

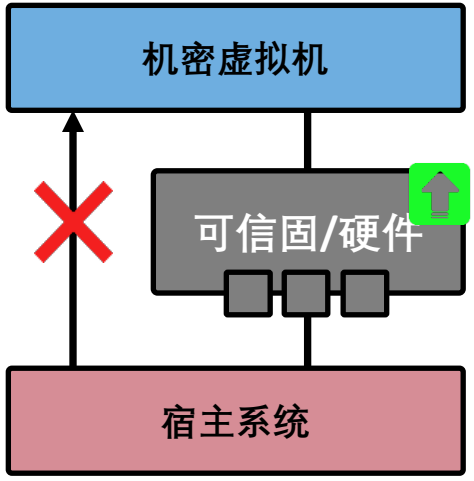

Alibaba Cloud
Azure
云厂商宿主系统
aws
腾讯云
Google Cloud


安全扫描、资源回收、
日志、监控、备份
快照、（热）迁
移、灾难恢复


机密虚拟机

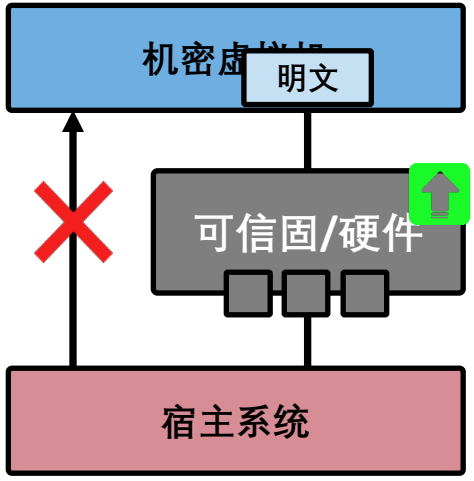
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入



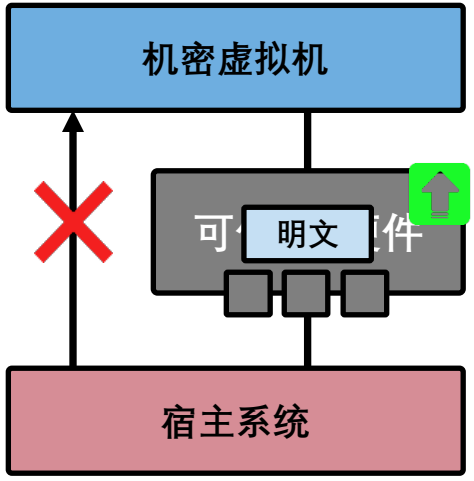
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入



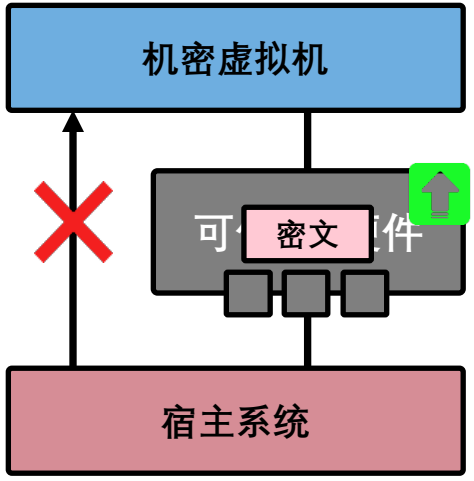
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入



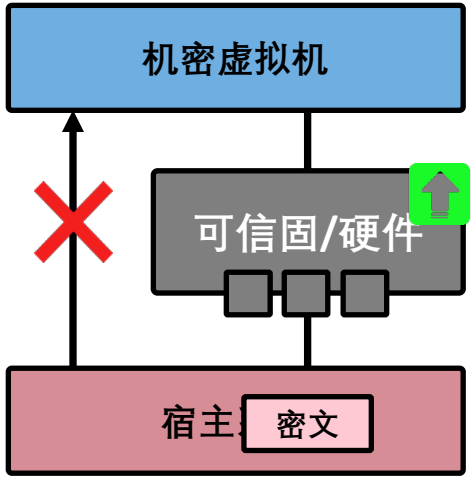
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入



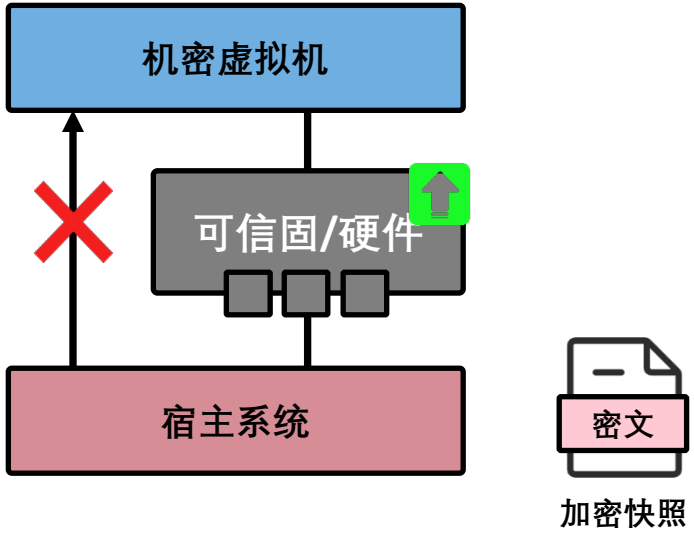
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入



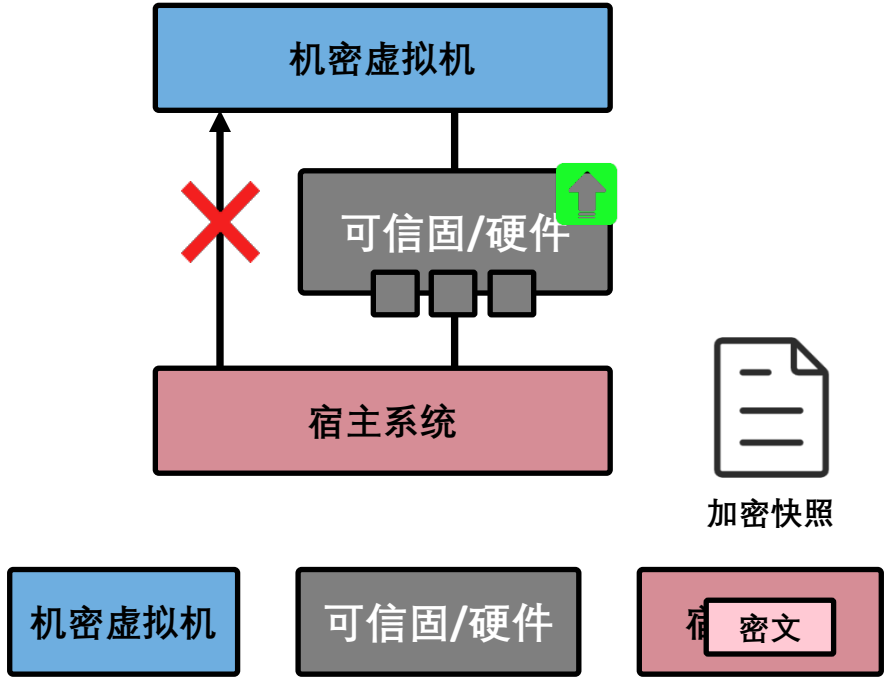
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入
  - 宿主机充当密文的搬运工
  - 快照、（热）迁移



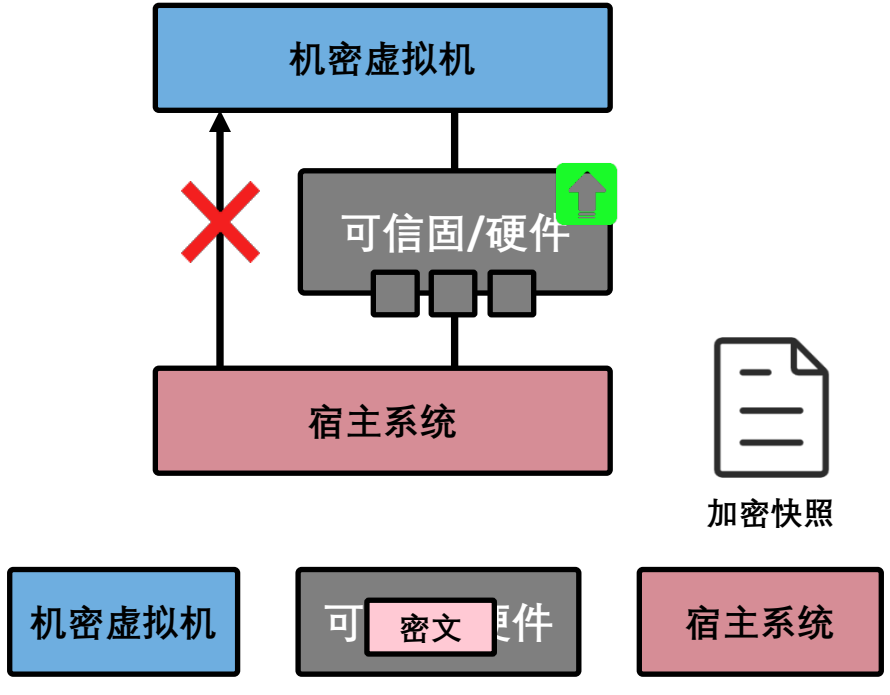
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入
  - 宿主机充当密文的搬运工
  - 快照、（热）迁移



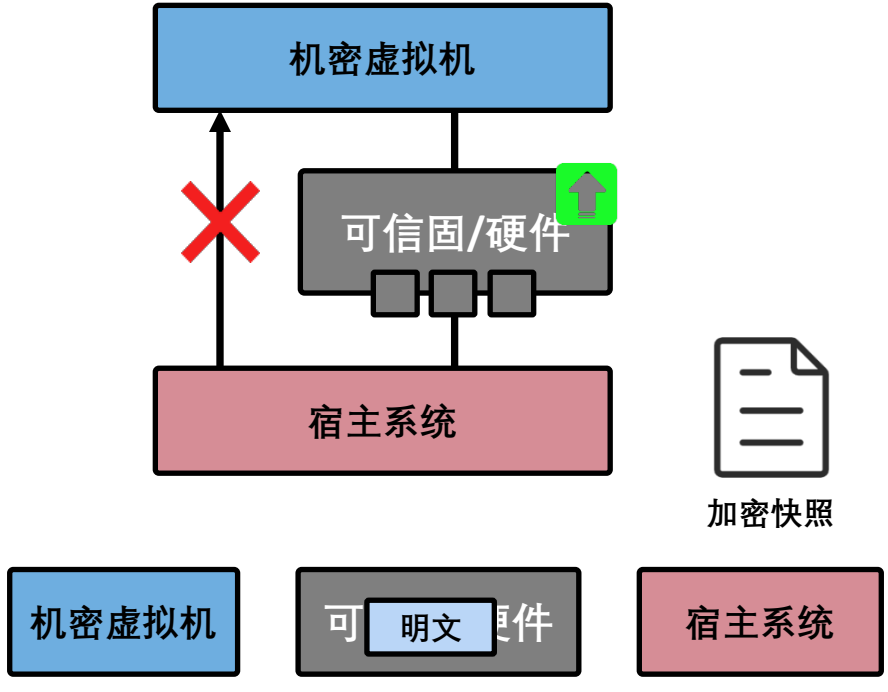
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入
  - 宿主机充当密文的搬运工
  - 快照、（热）迁移



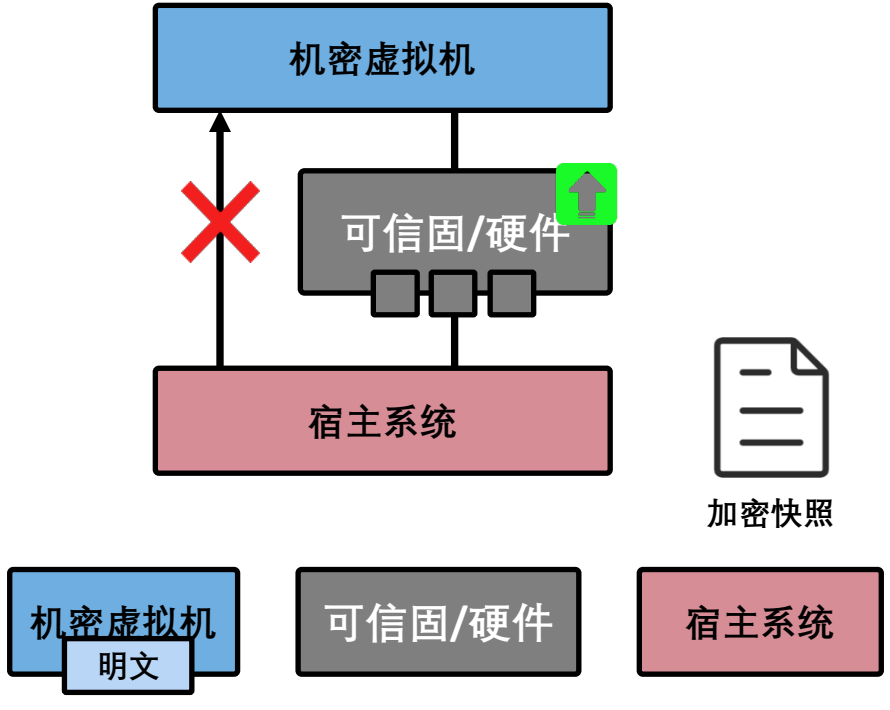
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入
  - 宿主机充当密文的搬运工
  - 快照、（热）迁移



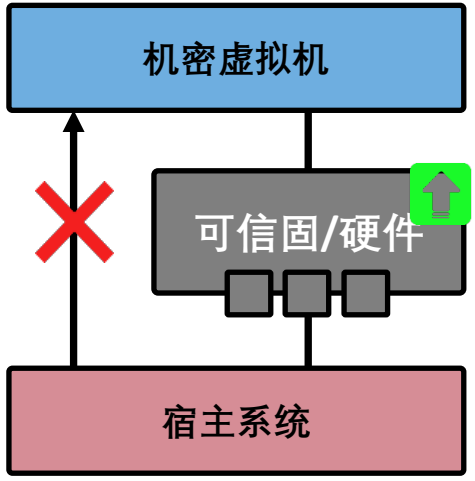
当前补救方案依赖硬件厂商

- 升级安全固件向宿主系统开放新的接口
  - 私有内存加密提取&解密注入
  - 状态加密提取&解密注入
  - 宿主机充当密文的搬运工
  - 快照、（热）迁移



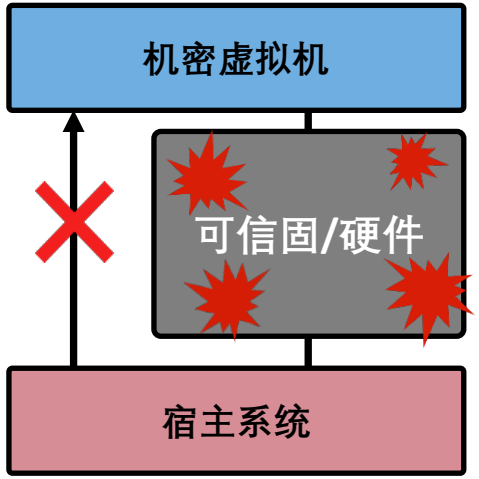
当前方案的缺陷

- 运维操作不灵活
  - 硬件厂商更新缓慢
  - 固件升级需要重启宿主机
  - 平台间运维进度参差不齐



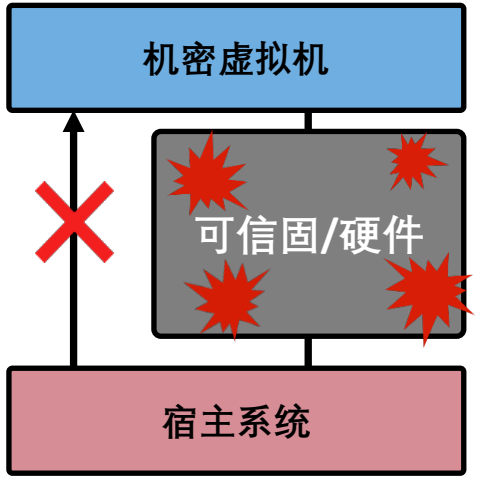
当前方案的缺陷

- 运维操作不灵活
  - 硬件厂商更新缓慢
  - 固件升级需要重启宿主机
  - 平台间运维进度参差不齐
- 降低CVM方案安全性
  - 可信固件TCB膨胀
  - 客户机和宿主机都以固/硬件为TCB



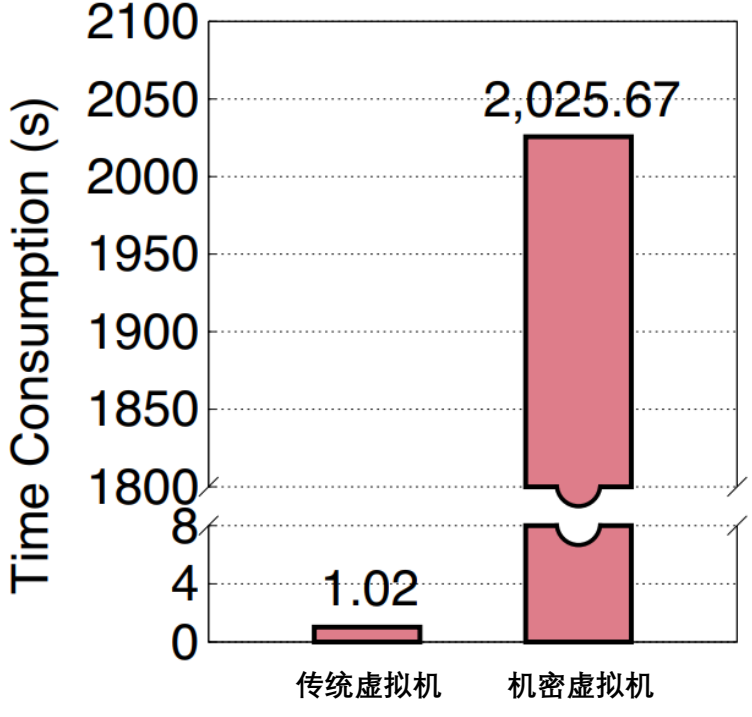
当前方案的缺陷

- 运维操作不灵活
  - 硬件厂商更新缓慢
  - 固件升级需要重启宿主机
  - 平台间运维进度参差不齐
- 降低CVM方案安全性
  - 可信固件TCB膨胀
  - 客户机和宿主机都以固/硬件为TCB
- 有性能问题



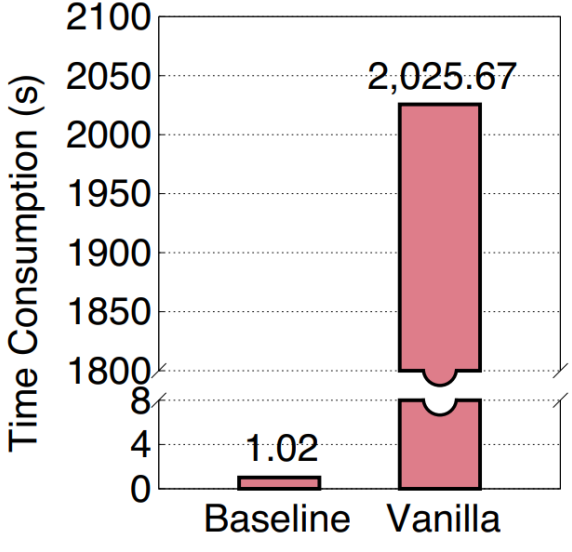
当前方案的缺陷

- AMD SEV官方热迁移方案耗时慢了**1986倍?**
  - 普通虚拟机1.02秒
  - 机密虚拟机2,025.67秒
- 测试配置
  - 128核心的AMD服务器
  - 客户机内存2GB，单vCPU
  - 源实例和目标实例在同一物理机上以避免不稳定网络的干扰

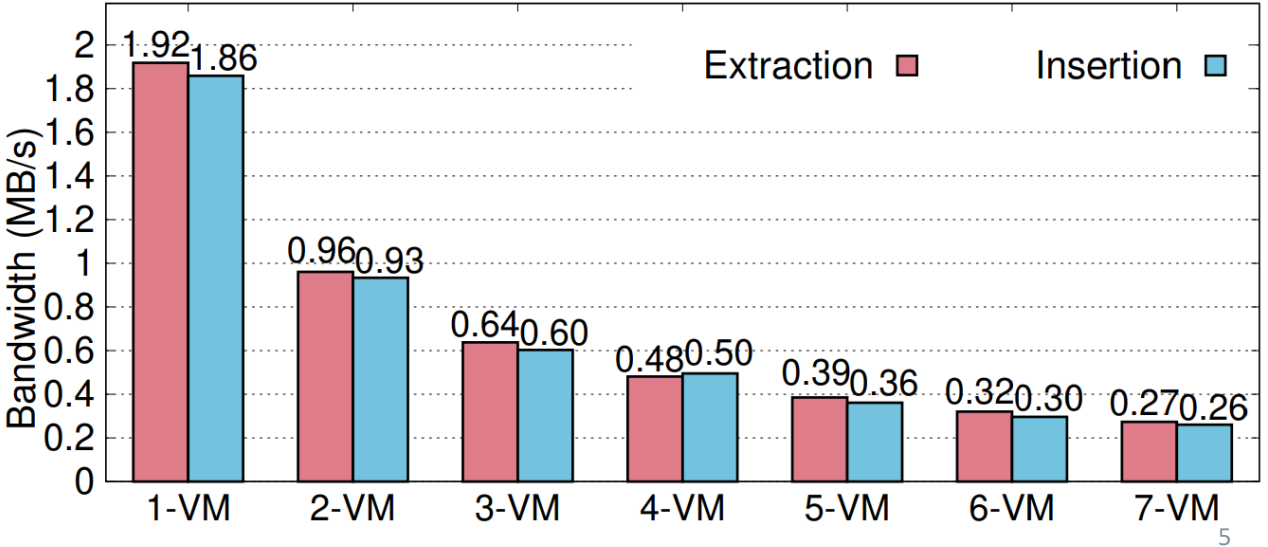


当前方案的缺陷

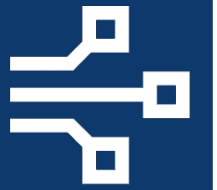
- AMD SEV官方热迁移方案耗时慢了**1986倍?**
  - 普通虚拟机1.02秒
  - 机密虚拟机2,025.67秒



- 加解密CVM内部数据的安全固件运行在了隔离的**AMD-SP**辅助核心上
  - 32位ARM核心，算力有限，**1.92MB/s**
  - 被所有CVM平分算力!

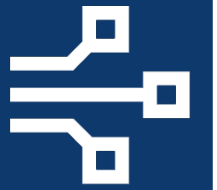


目标



- 更加**灵活**
  - 能让云厂商和云租户灵活定制运维操作
  - 更新升级运维模块时不需要重启整个物理机或是挂起/迁移实例
  - 兼容当前所有CVM硬件平台

目标



- 更加**灵活**
  - 能让云厂商和云租户灵活定制运维操作
  - 更新升级运维模块时不需要重启整个物理机或是挂起/迁移实例
  - 兼容当前所有CVM硬件平台

- 维护**安全**
  - CVM的安全能力不能降低，不能引入新的攻击方法

目标

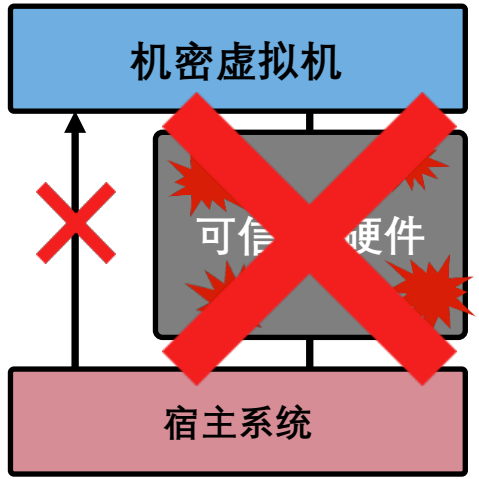
- 更加**灵活**
  - 能让云厂商和云租户灵活定制运维操作
  - 更新升级运维模块时不需要重启整个物理机或是挂起/迁移实例
  - 兼容当前所有CVM硬件平台

- 维护**安全**
  - CVM的安全能力不能降低，不能引入新的攻击方法

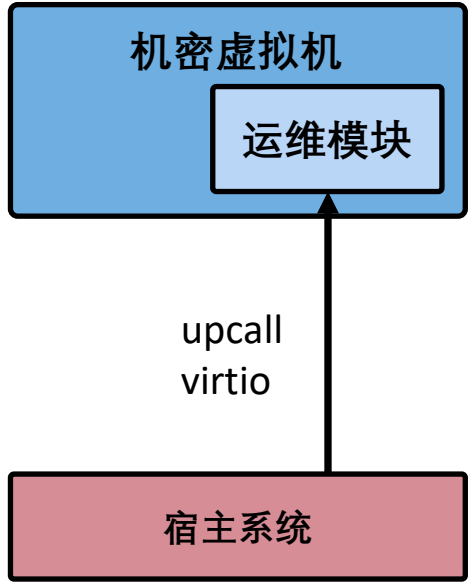
- 改善**性能**
  - 关键运维操作不受客户工作负载干扰
  - 不受各平台安全固件的性能限制，如AMD-SP

根本原因

运维模块放错了位置

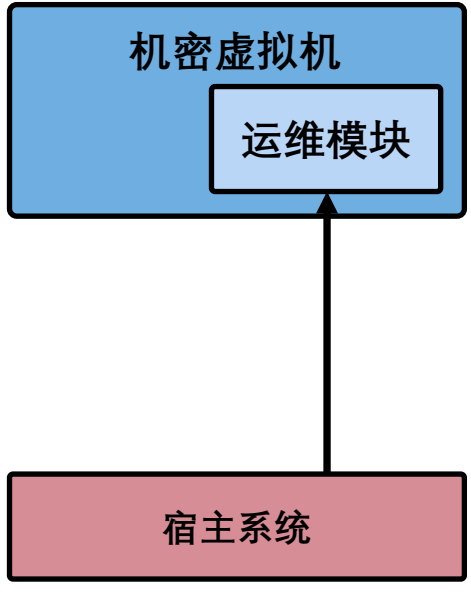


新解法



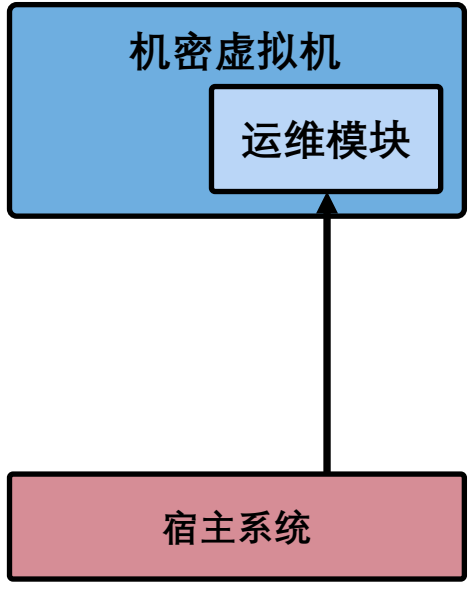
新解法

- 运维模块会与客户工作负载产生显著的算力竞争
  - 资源回收场景下耗时3倍



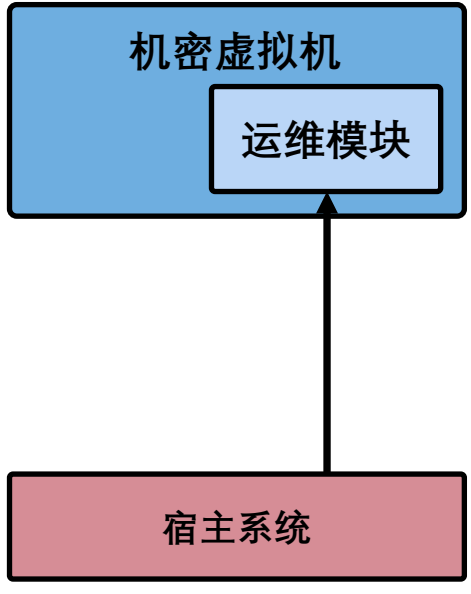
新解法

- 运维模块会与客户工作负载产生显著的算力竞争
  - 资源回收场景下耗时3倍
- 运维模块无法防御客户系统崩溃
  - 灾难恢复、错误诊断等操作需要能够在客户系统出错的环境下继续正确运行



新解法

- 运维模块会与客户工作负载产生显著的算力竞争
  - 资源回收场景下耗时3倍
- 运维模块无法防御客户系统崩溃
  - 灾难恢复、错误诊断等操作需要能够在客户系统出错的环境下继续正确运行
- 需要一种机制提供新的语义：
  - 宿主用**独立**且**受保护**的资源来调用目标运维模块



观察和新Idea

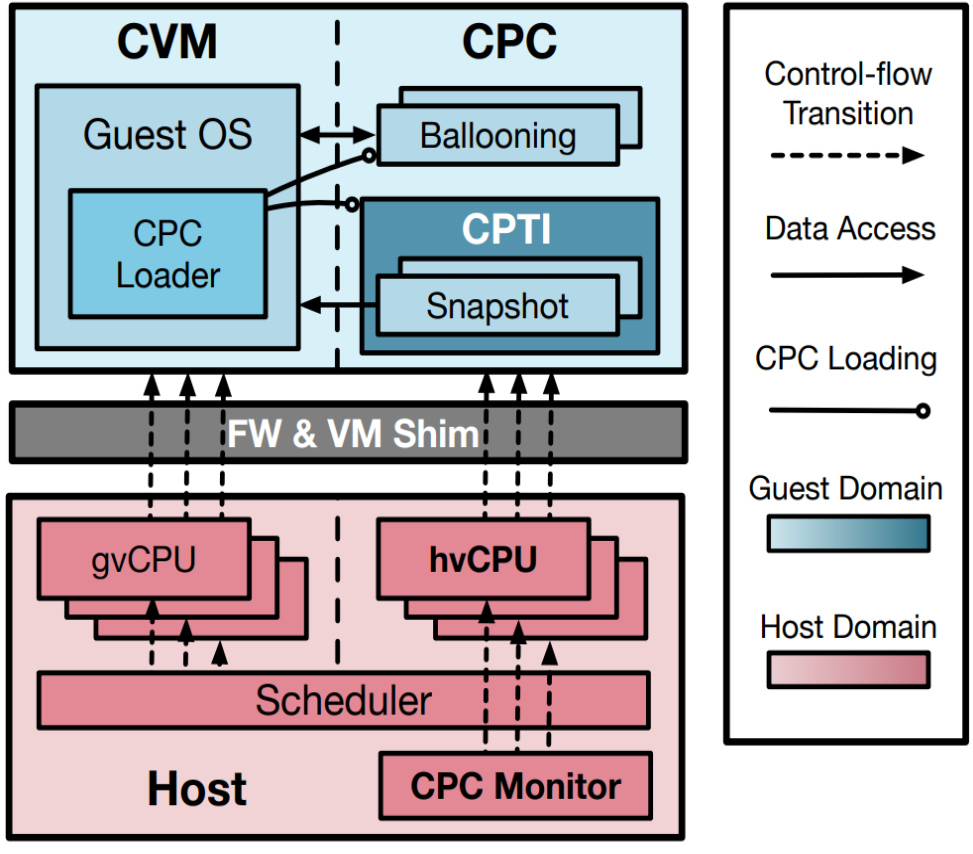
- CVM的确有效阻断了宿主系统对客户机**数据面**的访问
  - 但是依然保留了宿主系统对客户机部分**控制面**的影响
  - 例如宿主系统依然可以调度或睡眠CVM的vCPU线程

将宿主系统**调度vCPU线程**的语义扩展为**调用运维过程**的语义


CPC
onfidential
rocedure
alls

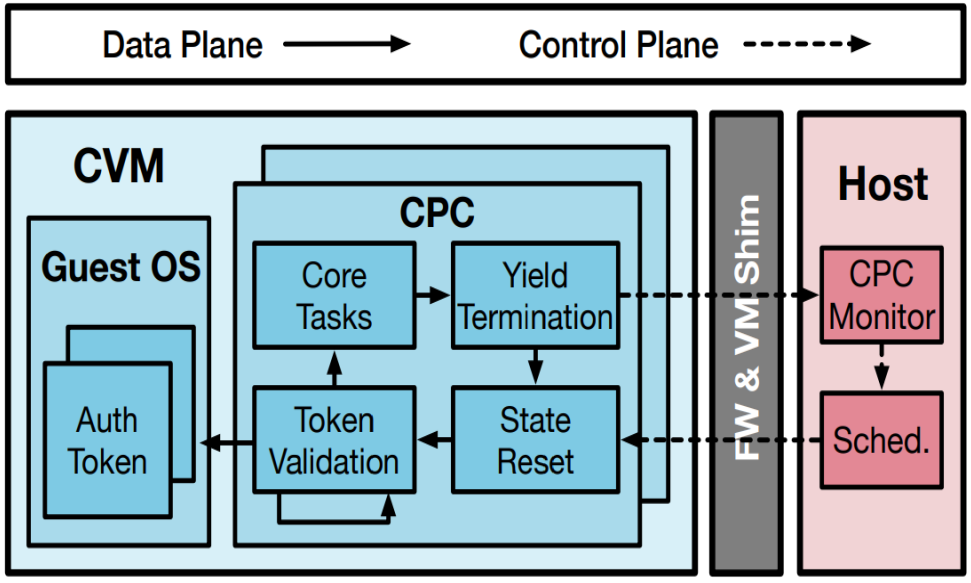
机密过程调用CPC

- vCPU分为**gvCPU**和**hvCPU**
  - **gvCPU**和传统vCPU一样参与宿主机线程调度
  - **hvCPU**不参与宿主机线程调度，被绑定了运维模块，只在宿主机需要对CVM进行运维操作时才被从睡眠队列中唤醒调度
    - 客户机操作系统逻辑上“看不见”hvCPU



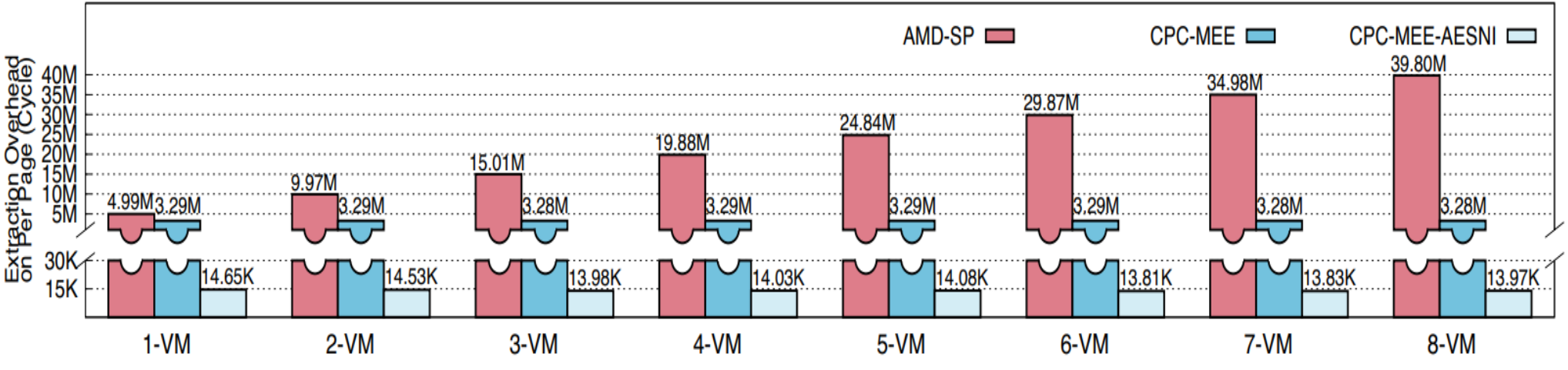
CPC状态机

- **客户数据面**与**宿主控制面**共同驱动的状态机
  - 当宿主系统需要调用某个运维功能时，就直接唤醒并调度相应的hvCPU
  - 验证授权保护客户机安全
  - 无限循环，可以多次被调用
- 维护了客户机与宿主系统间**清晰的安全边界**
- 复用了现有**成熟的机制**
  - 线程调度和hypercall



基于CPC的Snapshot性能测试

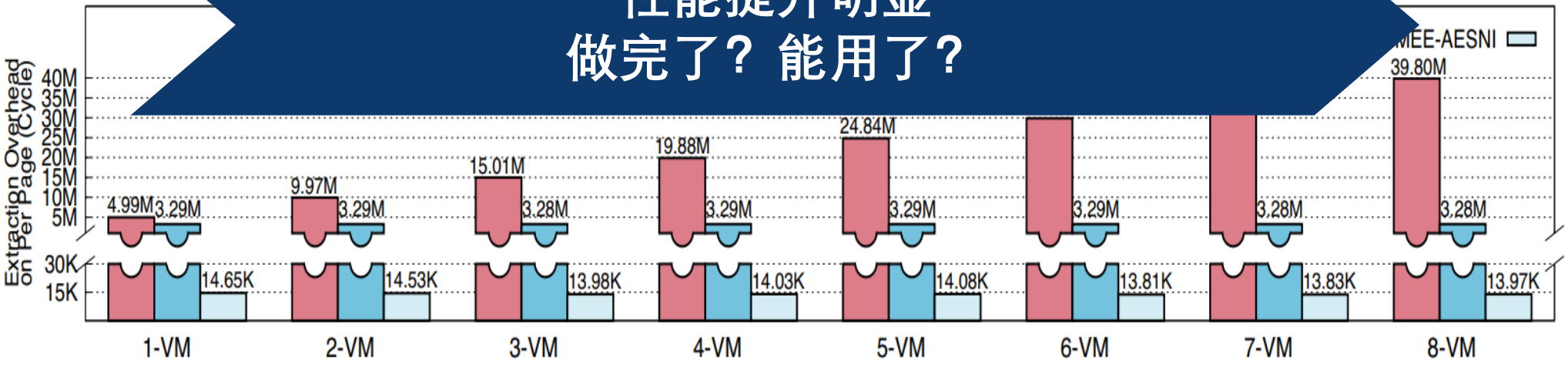
- 自行软件实现的CPC-Snapshot：
  - 1-VM下开销降低34%
  - 8-VM下加速12倍，VM越多提升越大
  - 优秀的Scalability
- 开启AESNI加速的CPC-Snapshot：
  - 1-VM下提速341倍
  - 8-VM下加速2849倍，VM越多提升越大
  - 依然优秀的Scalability



基于CPC的Snapshot性能测试

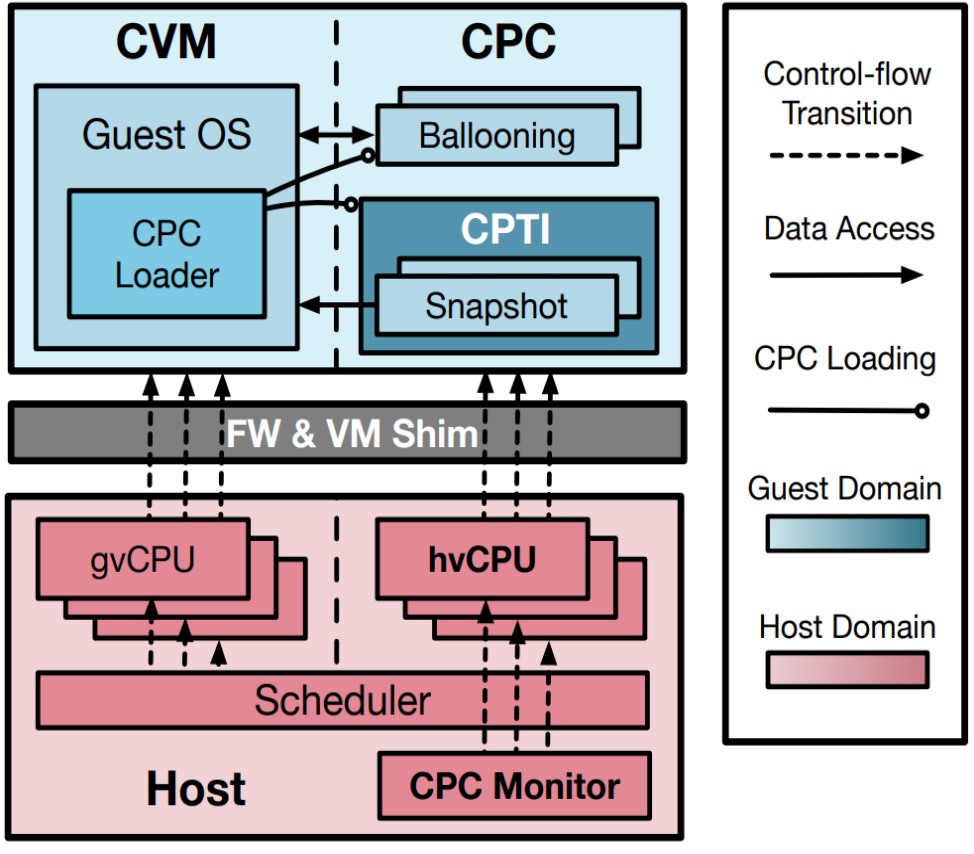
- 自行软件实现的CPC-Snapshot：
  - 1-VM下开销降低34%
  - 8-VM下加速12倍，VM越多提升越大
  - [illegible]
- 开启AESNI加速的CPC-Snapshot：
  - 1-VM下提速341倍
  - 8-VM下加速2849倍，VM越多提升越大
  - [illegible]

性能提升明显
做完了？能用了？



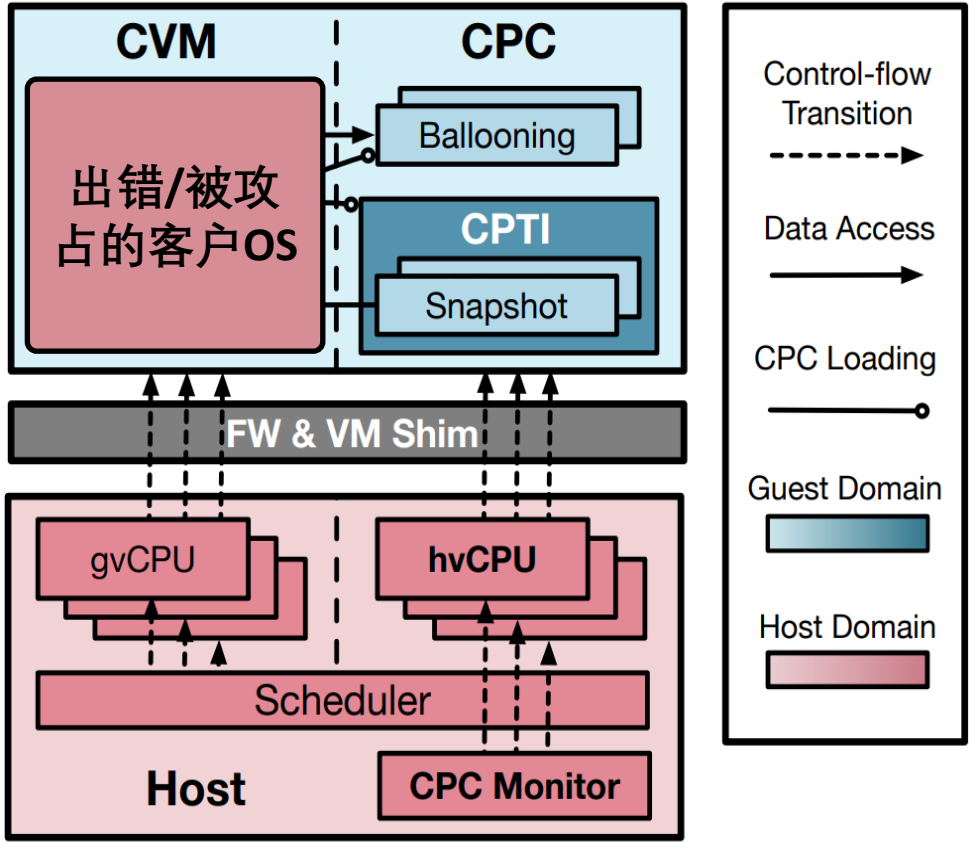
比CVM更艰巨的威胁模型

- 部分运维操作要在客户机崩溃场景下继续正确运行
  - 例如，灾难恢复，数据备份，错误分析



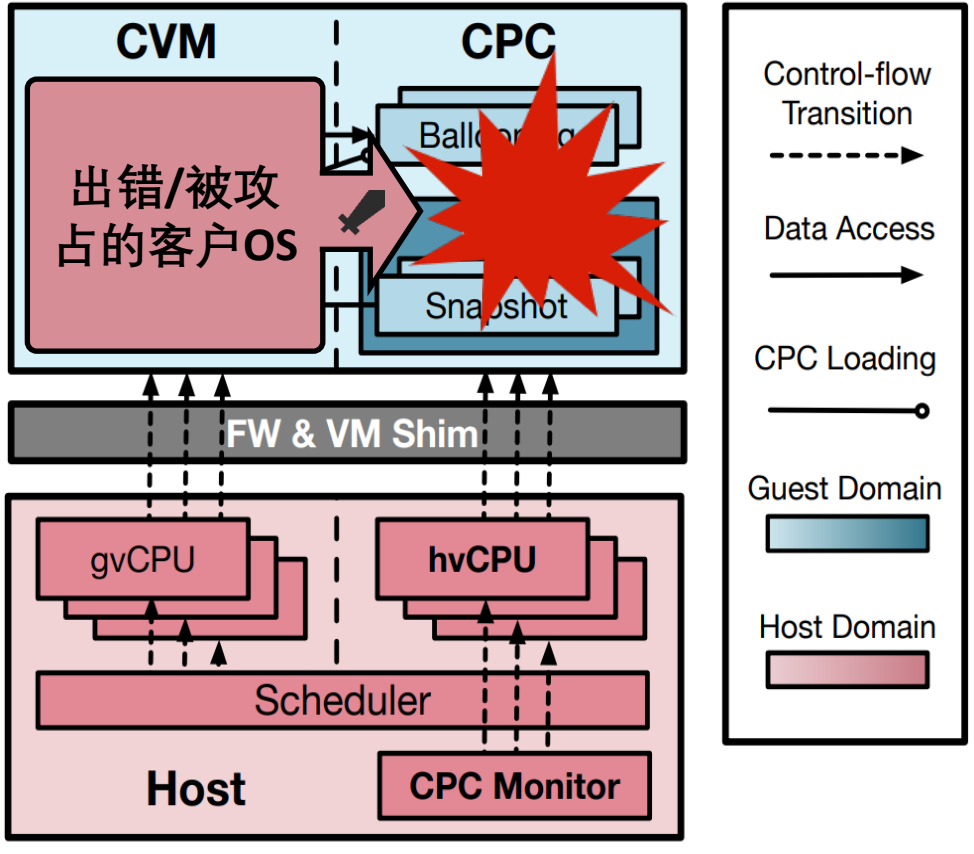
比CVM更艰巨的威胁模型

- 部分运维操作要在客户机崩溃场景下继续正确运行
  - 例如，灾难恢复，数据备份，错误分析
- 客户机的OS通常TCB巨大
  - 通常为Linux
  - 漏洞多，易崩溃，不安全



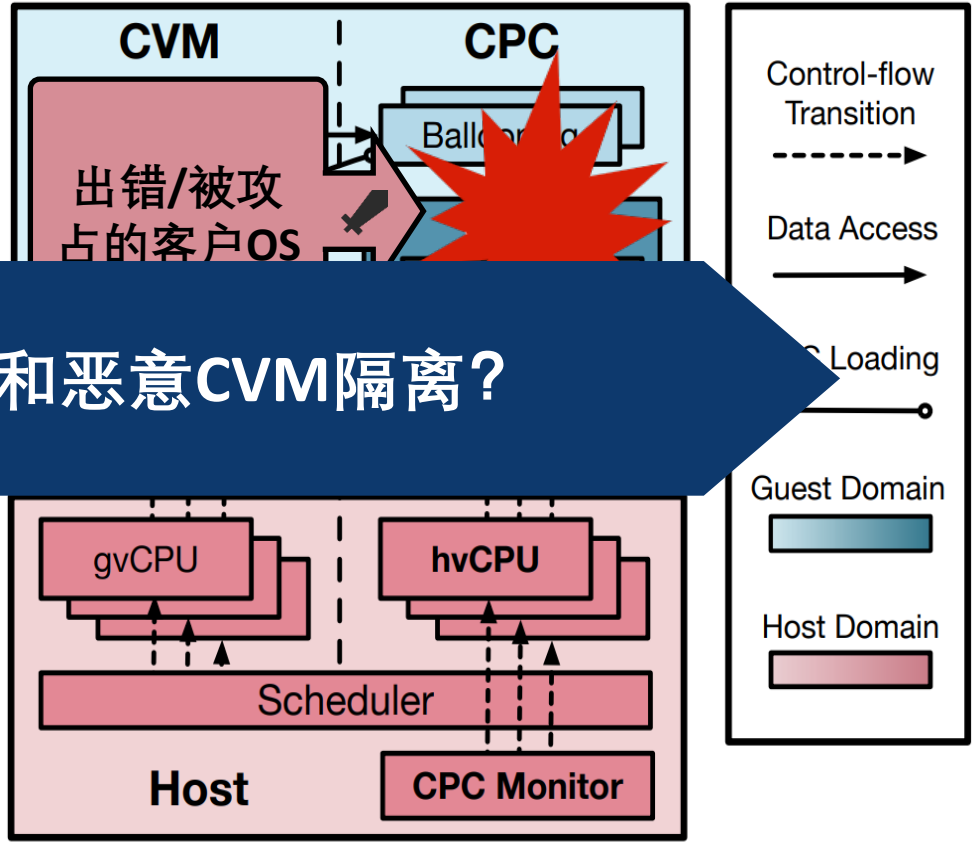
比CVM更艰巨的威胁模型

- 部分运维操作要在客户机崩溃场景下继续正确运行
  - 例如，灾难恢复，数据备份，错误分析
- 客户机的OS通常TCB巨大
  - 通常为Linux
  - 漏洞多，易崩溃，不安全
- 当前CPC可能被出错（或被攻占）的客户OS篡改
  - 导致对应CPC功能损坏，重要数据无法被抢救或是恢复



比CVM更艰巨的威胁模型

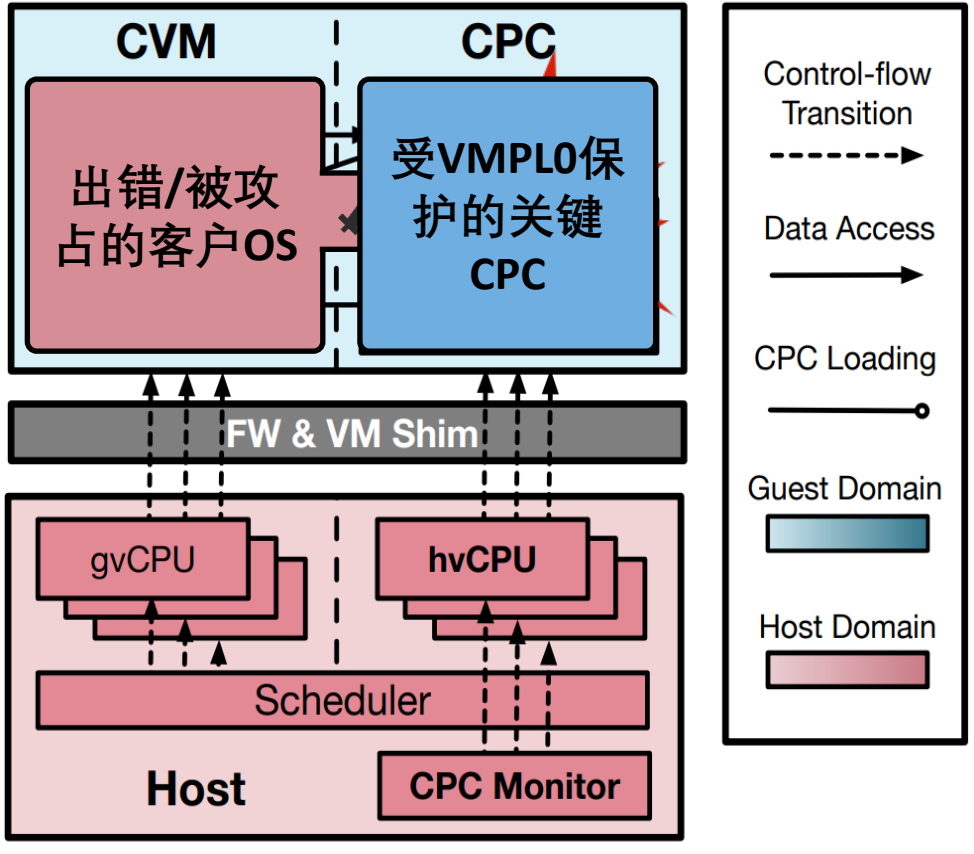
- 部分运维操作要在客户机崩溃场景下继续正确运行
  - 例如，灾难恢复，数据备份，错误分析
- 客户
  - 通
  - 漏
- 当前CPC可能被出错（或被攻占）的客户OS篡改
  - 导致对应CPC功能损坏，重要数据无法被抢救或是恢复



如何让关键CPC能和恶意CVM隔离？

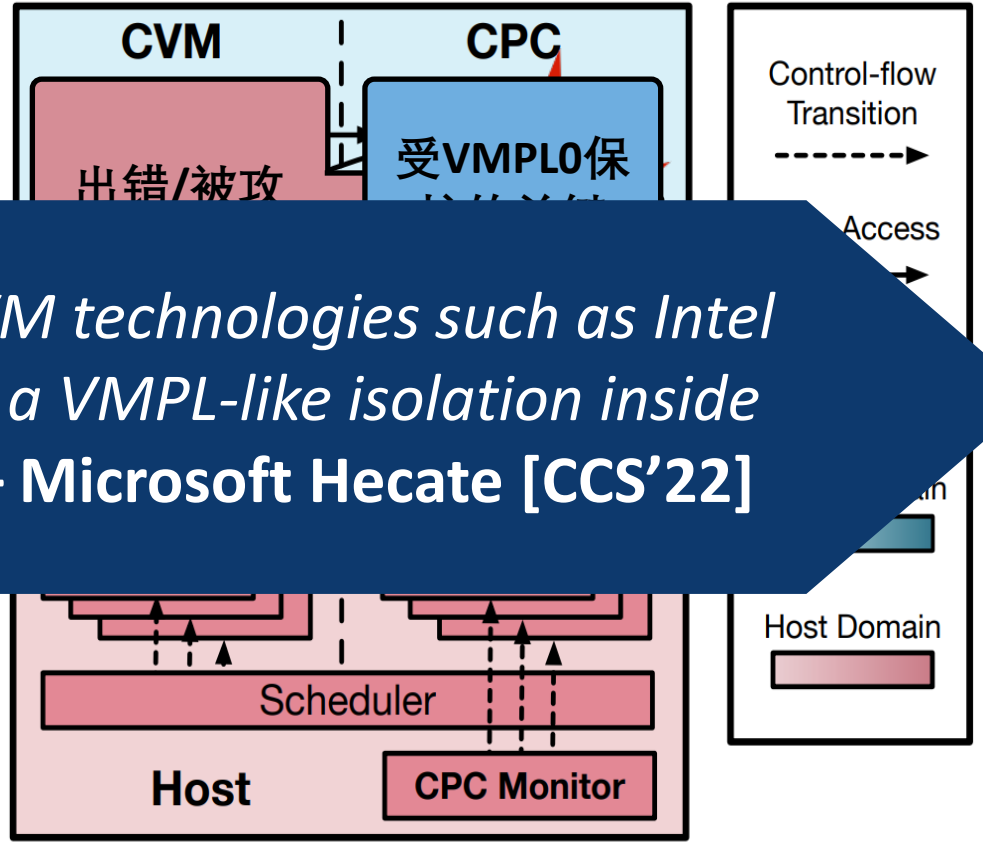
比CVM更艰巨的威胁模型

- AMD SEV平台有VMPL特性
  - 将客户OS运行在VMPL1-3（低特权）
  - 将关键CPC运行在VMPL0（高特权）
  - 客户OS无法访问VMPL0的内存和寄存器
- 微软的Hecate[CCS22]已经利用该硬件特性向AMD CVM提供安全服务
  - 防火墙，兼容性层等



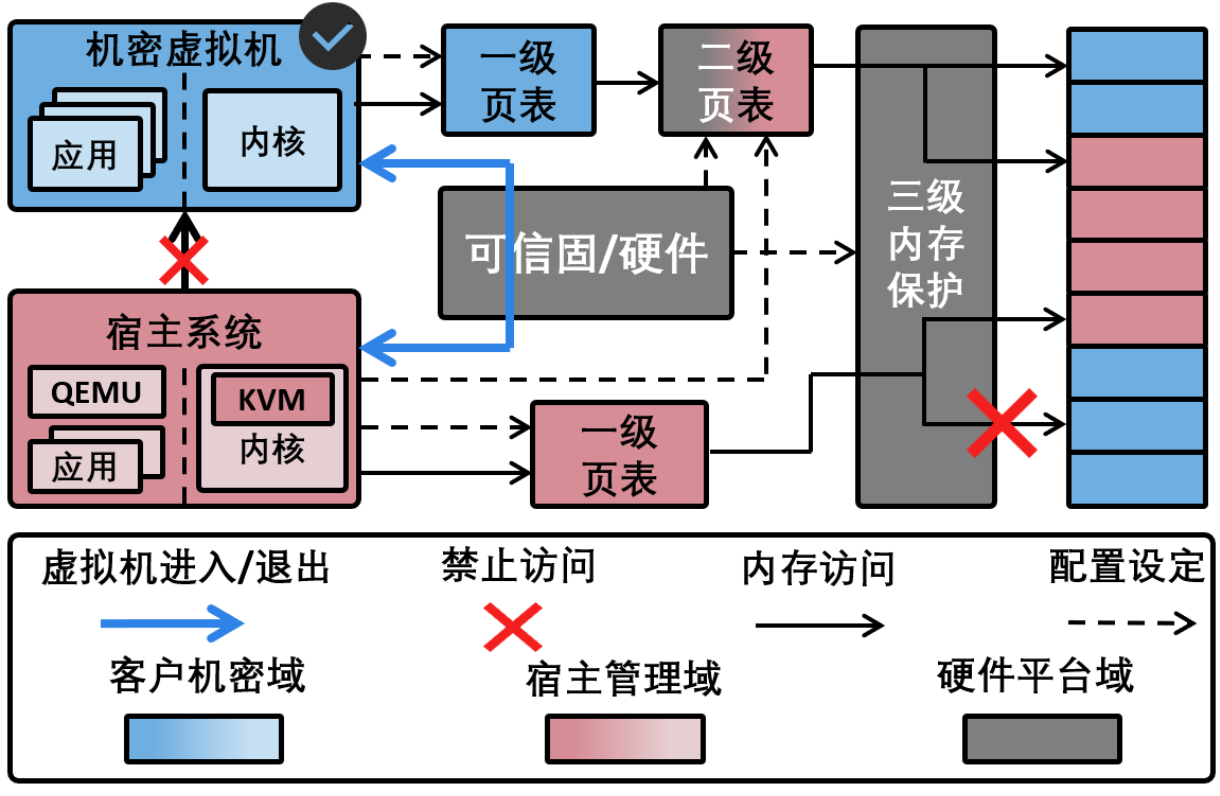
比CVM更艰巨的威胁模型

- AMD SEV平台有VMPL特性
  - 将客户OS运行在VMPL1-3（低特权）
  - 将关键CPC运行在VMPL0（高特权）
- 微
  特性

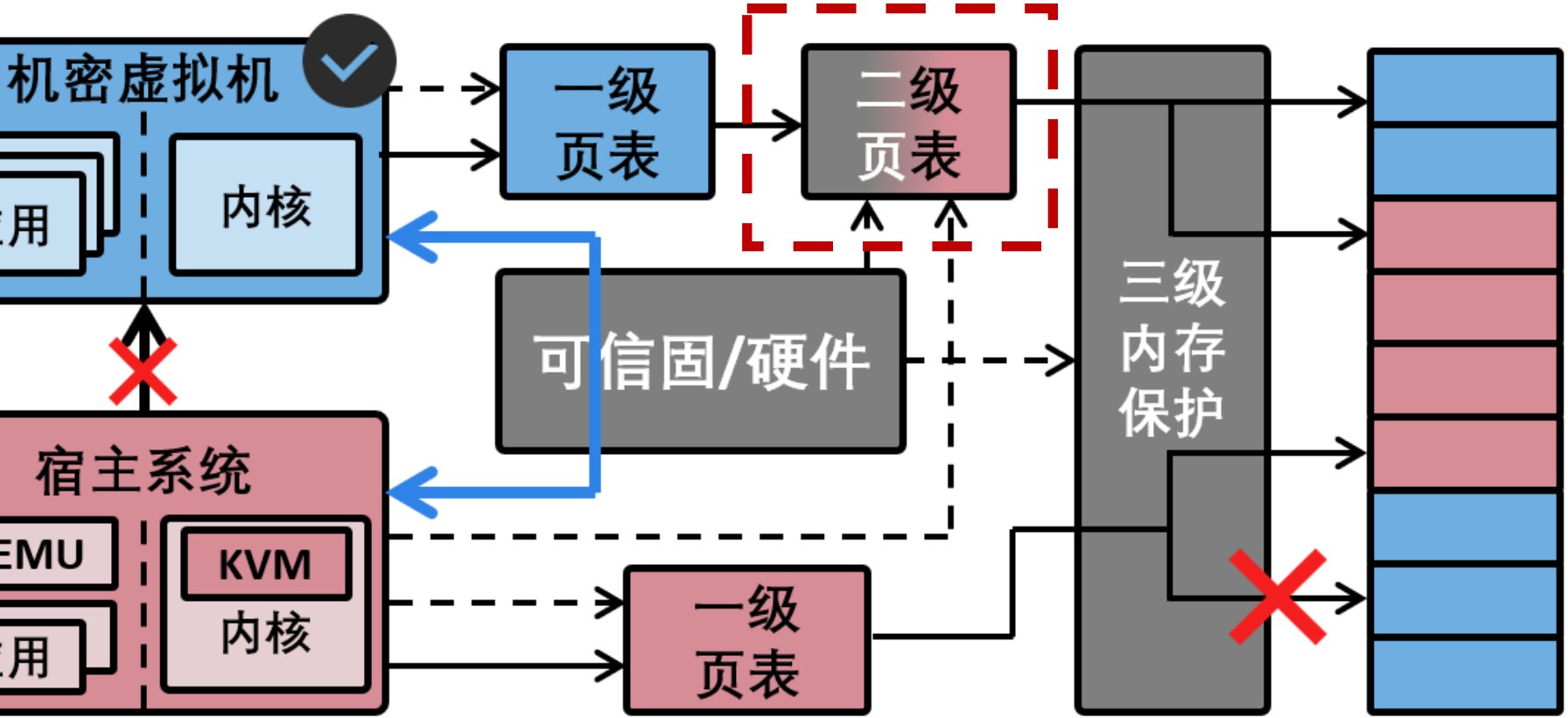


> *"Other new confidential VM technologies such as Intel TDX and ARM Realm lack a VMPL-like isolation inside their confidential VMs."* – **Microsoft Hecate [CCS'22]**

只能给AMD用了吗?


机密虚拟机
应用
内核
一级
页表
二级
页表
三级
内存
保护
可信固/硬件
宿主系统
QEMU
应用
KVM
内核
一级
页表
虚拟机进入/退出
禁止访问
内存访问
配置设定
客户机密域
宿主管理域
硬件平台域

求同存异



求同存异

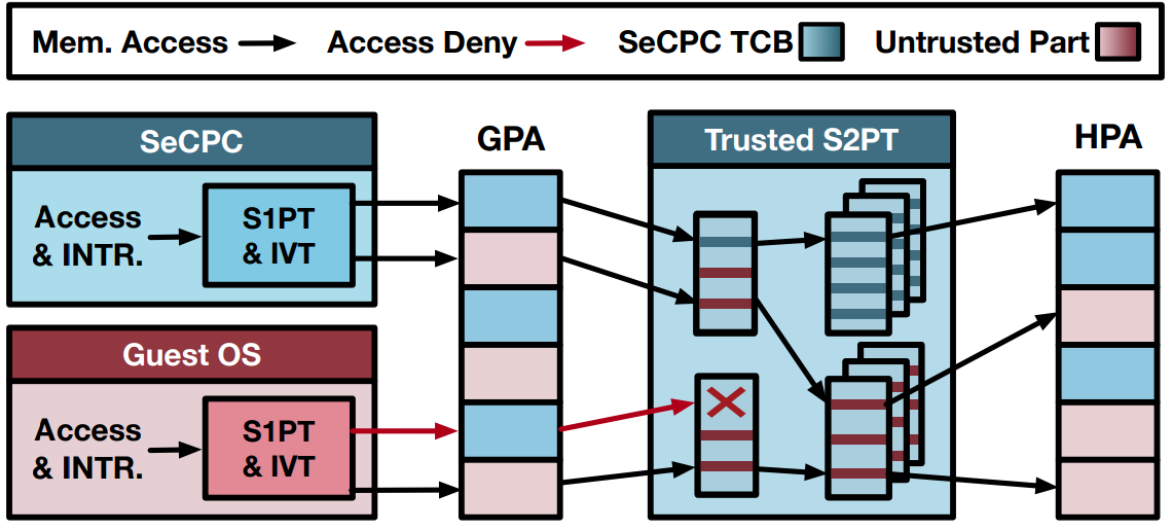
- AMD SEV平台**有VMPL特性**
  - 但CVM私有内存的**二级页表由宿主系统管理，不可信**

- Intel TDX、ARM CCA、RISC-V CoVE平台**没有VMPL特性**
  - 但CVM私有内存的**二级页表由可信固件管理，可信!**

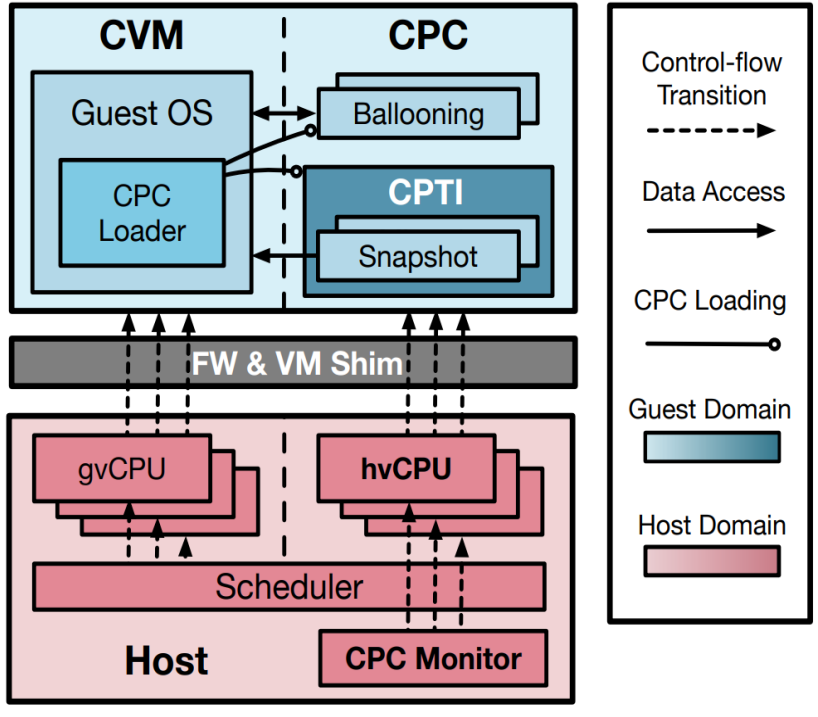
机密页表隔离（CPTI）

- Confidential Page Table Isolation，简称CPTI
  - CPC有了CPTI的帮助就进化成了**SeCPC (Secure CPC)**

- 安全固件为关键CPC的hvCPU创建独立的**可信二级页表**
  - SeCPC的二级页表中不仅有客户机的内存，还映射了额外**可信内存**
  - 客户OS的gvCPU的二级页表中没有这些映射
- SeCPC进一步在可信内存中构建可信**一级页表**和**中断向量表**

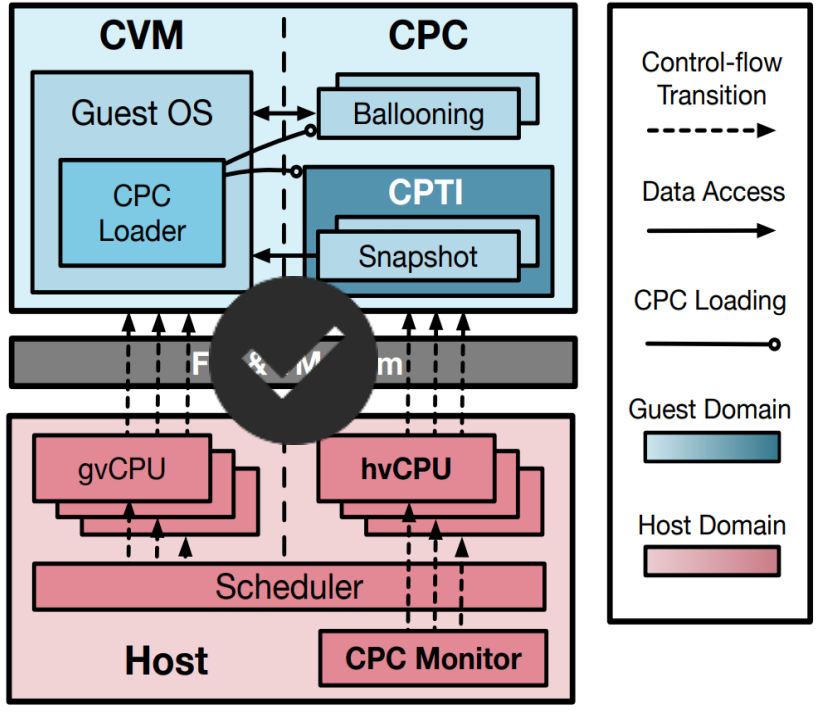


安全分析（ARM CCA）


CVM
CPC
Guest OS
Ballooning
CPC Loader
CPTI
Snapshot
FW & VM Shim
gvCPU
hvCPU
Scheduler
Host
CPC Monitor
Control-flow Transition
Data Access
CPC Loading
Guest Domain
Host Domain

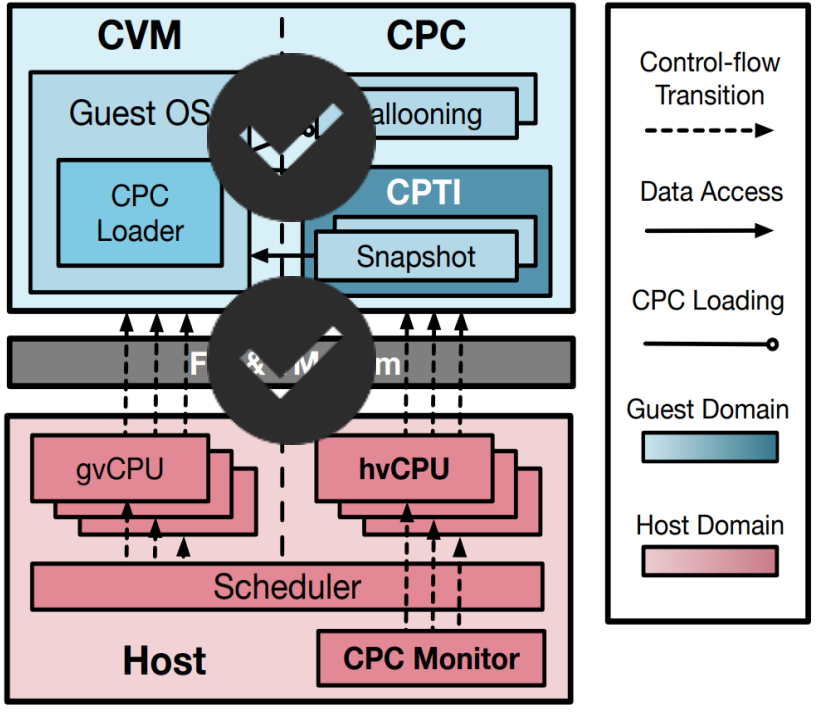
安全分析（ARM CCA）

- 更精简的硬/固件
  - 两个简单的运维模块（快照和安全日志）比CPTI的代码大了7.23倍



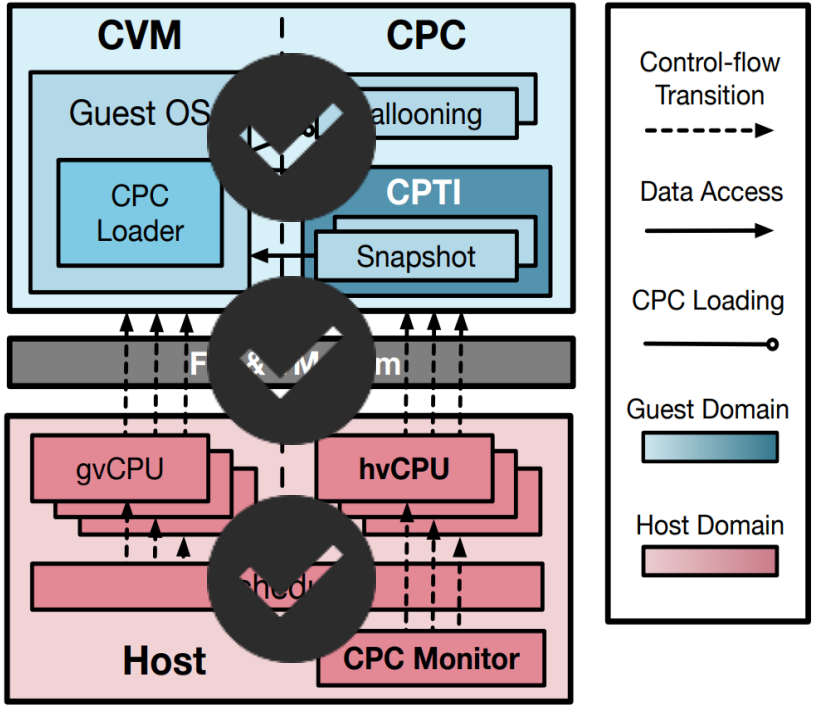
安全分析（ARM CCA）

- 更精简的硬/固件
  - 两个简单的运维模块（快照和安全日志）比CPTI的代码大了7.23倍
- 客户机安全性：
  - CPC代码量和客户Linux相比是小的
  - CPC能被更快更灵活地打上安全补丁
  - CPC一旦出错，CPTI也能用于隔离CPC
  - 客户只需装备真正需要的运维CPC



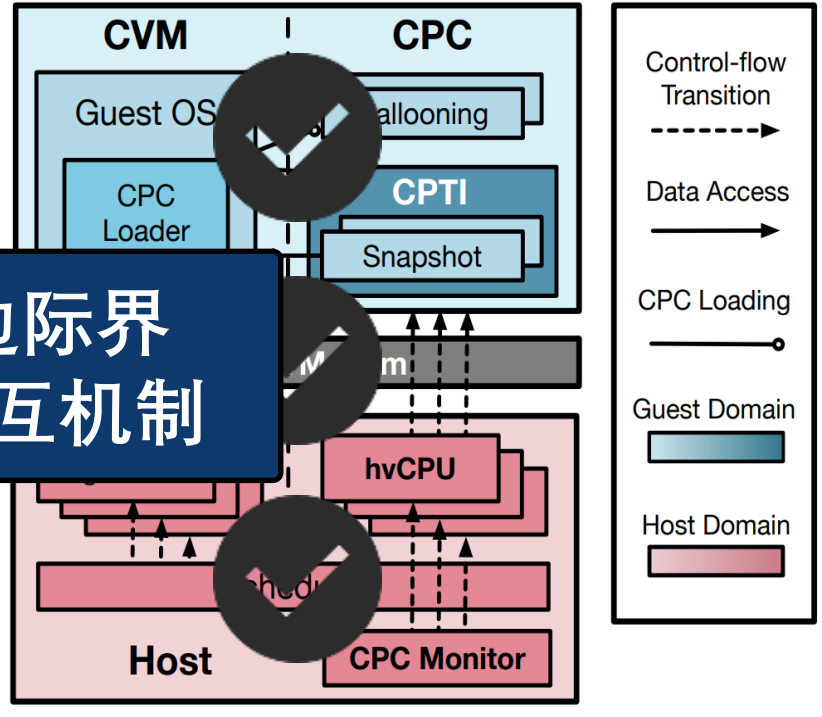
安全分析（ARM CCA）

- 更精简的硬/固件
  - 两个简单的运维模块（快照和安全日志）比CPTI的代码大了7.23倍
- 客户机安全性：
  - CPC代码量和客户Linux相比是小的
  - CPC能被更快更灵活地打上安全补丁
  - CPC一旦出错，CPTI也能用于隔离CPC
  - 客户只需装备真正需要的运维CPC
- 宿主机安全性：
  - KVM代码增长仅280行，大部分修改都在用户态QEMU和KVMTOOL中



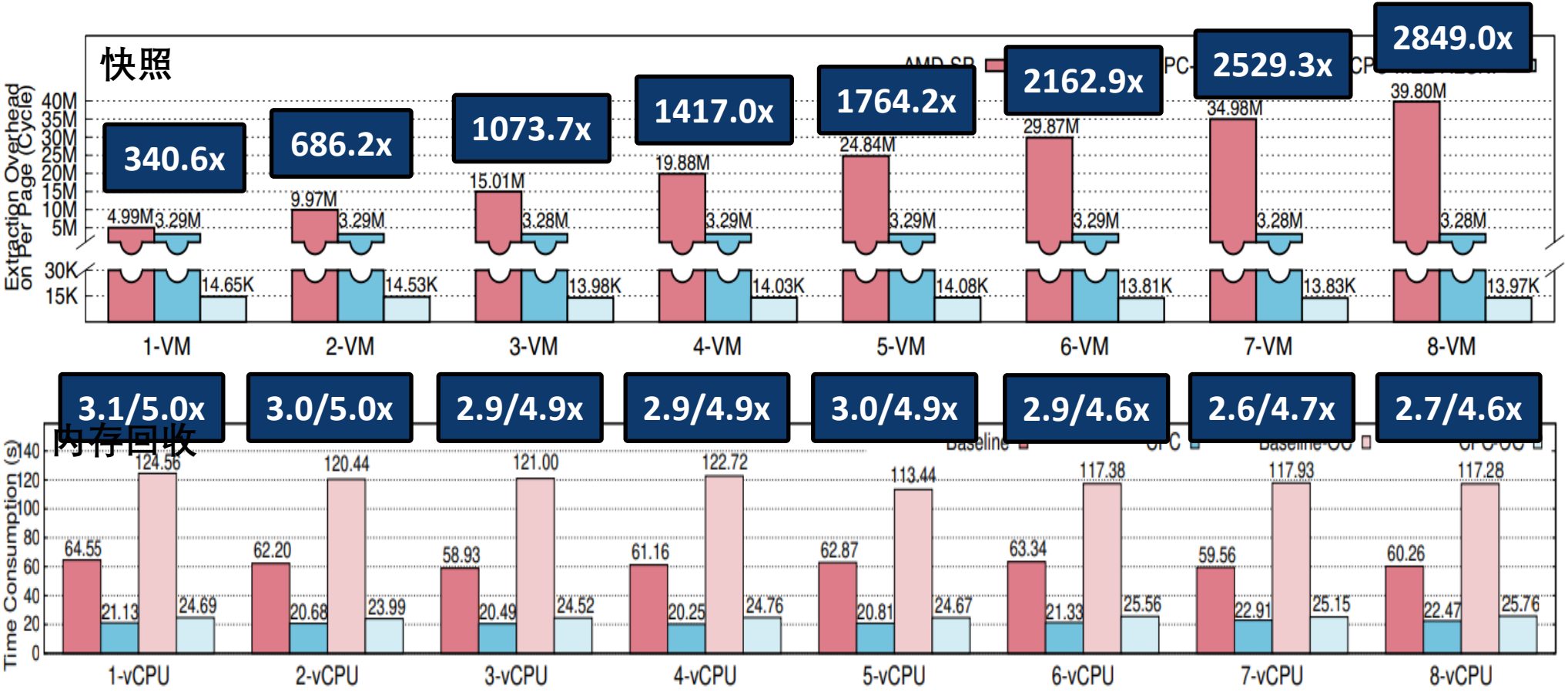
安全分析（ARM CCA）

- 更精简的硬/固件
  - 两个简单的运维模块（快照和安全日志）比CPTI的代码大了7.23倍
- 客户机安全性：
  - CPC代码量和客户Linu
  - CPC能被更快更灵活地
  - CPC一旦出错，CPTI也能用于隔离CPC
  - 客户只需装备真正需要的运维CPC
- 宿主机安全性：
  - KVM代码增长仅280行，大部分修改都在用户态QEMU和KVMTOOL中



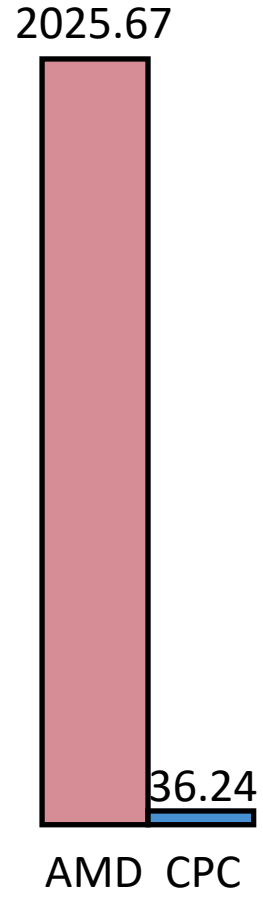
清晰的安全边际界
复用成熟的交互机制

性能测试（AMD SEV）


快照
Extraction Overhead on Per Page (Cycle)
340.6x
686.2x
1073.7x
1417.0x
1764.2x
2162.9x
2529.3x
2849.0x
4.99M
3.29M
14.65K
9.97M
3.29M
14.53K
15.01M
3.28M
13.98K
19.88M
3.29M
14.03K
24.84M
3.29M
14.08K
29.87M
3.29M
13.81K
34.98M
3.28M
13.83K
39.80M
3.28M
13.97K
1-VM
2-VM
3-VM
4-VM
5-VM
6-VM
7-VM
8-VM
内存回收
3.1/5.0x
3.0/5.0x
2.9/4.9x
2.9/4.9x
3.0/4.9x
2.9/4.6x
2.6/4.7x
2.7/4.6x
Time Consumption (s)
64.55
21.13
124.56
24.69
62.20
20.68
120.44
23.99
58.93
20.49
121.00
24.52
61.16
20.25
122.72
24.76
62.87
20.81
113.44
24.67
63.34
21.33
117.38
25.56
59.56
22.91
117.93
25.15
60.26
22.47
117.28
25.76
1-vCPU
2-vCPU
3-vCPU
4-vCPU
5-vCPU
6-vCPU
7-vCPU
8-vCPU

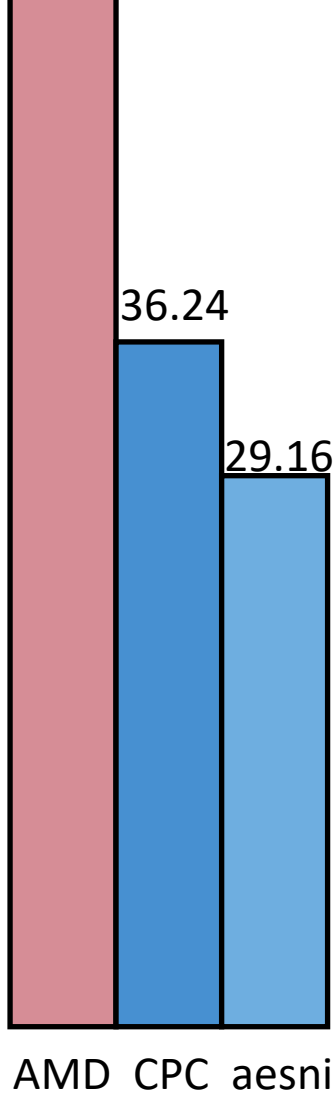
性能测试

- 热迁移场景对比AMD官方方案：
  - AES-GCM算法软件实现，加速达到**55.90倍**
    - 移植mbedtls



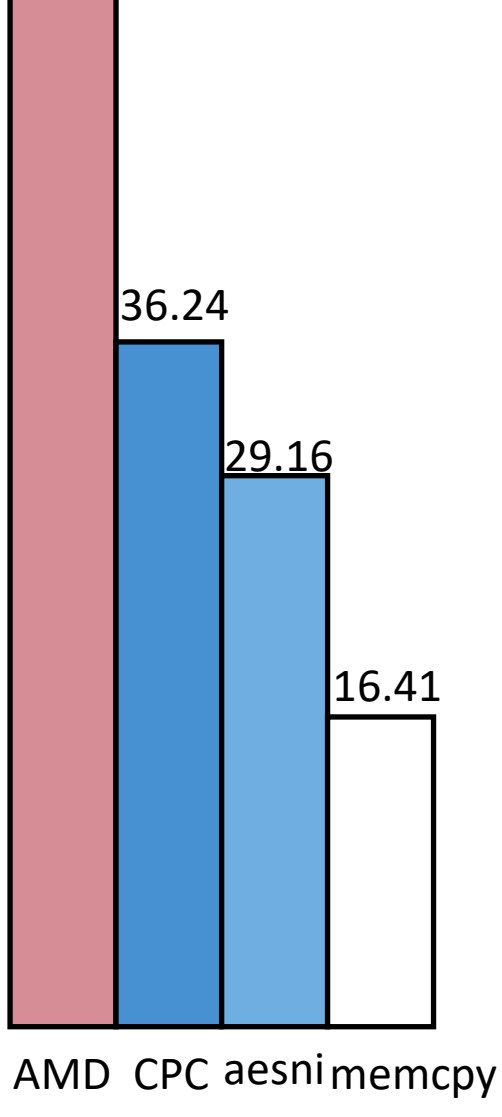
性能测试

- 热迁移场景对比AMD官方方案：
  - AES-GCM算法软件实现，加速达到**55.90倍**
    - 移植mbedtls
  - AESNI加速后，加速达到**69.47倍**
    - 现在大部分开销是GCM编码



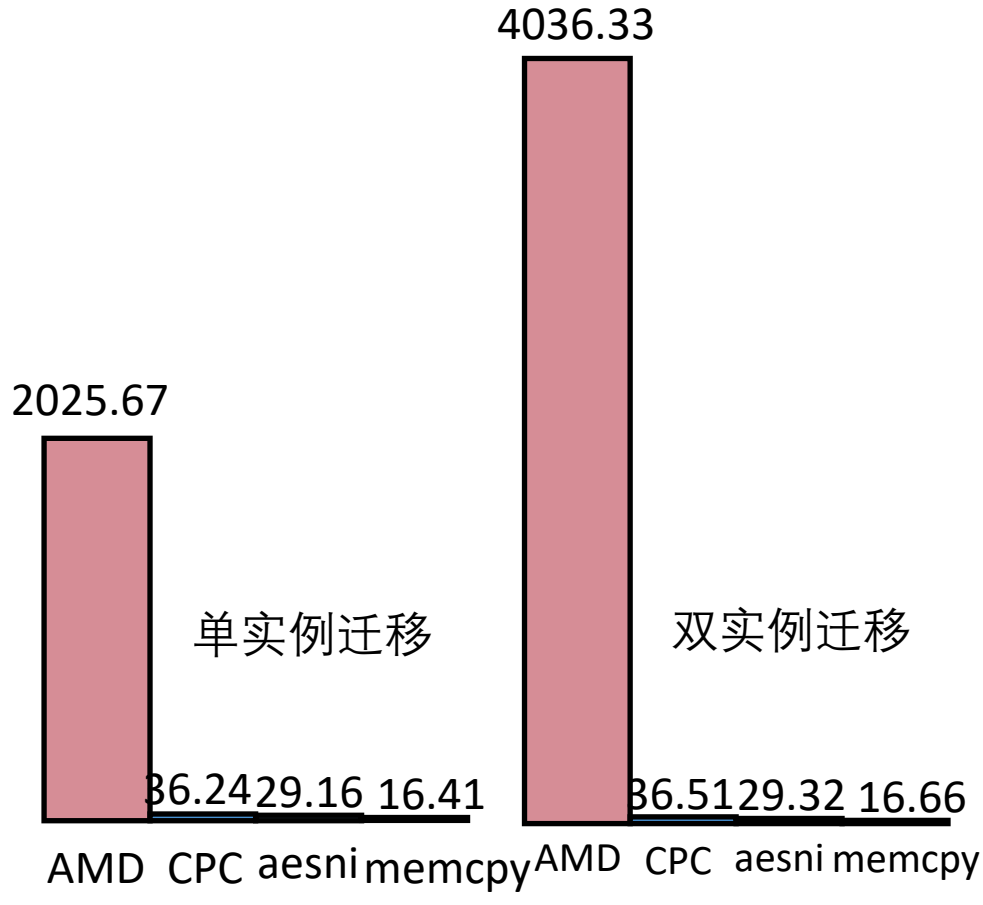
性能测试

- 热迁移场景对比AMD官方方案：
  - AES-GCM算法软件实现，加速达到**55.90倍**
    - 移植mbedtls
  - AESNI加速后，加速达到**69.47倍**
    - 现在大部分开销是GCM编码
  - 要是AES-GCM彻底被加速成普通的内存拷贝呢？
    - 用memcpy代替加密，直接拷贝出明文来模拟
    - 16.41秒，加速比为**123.44倍**



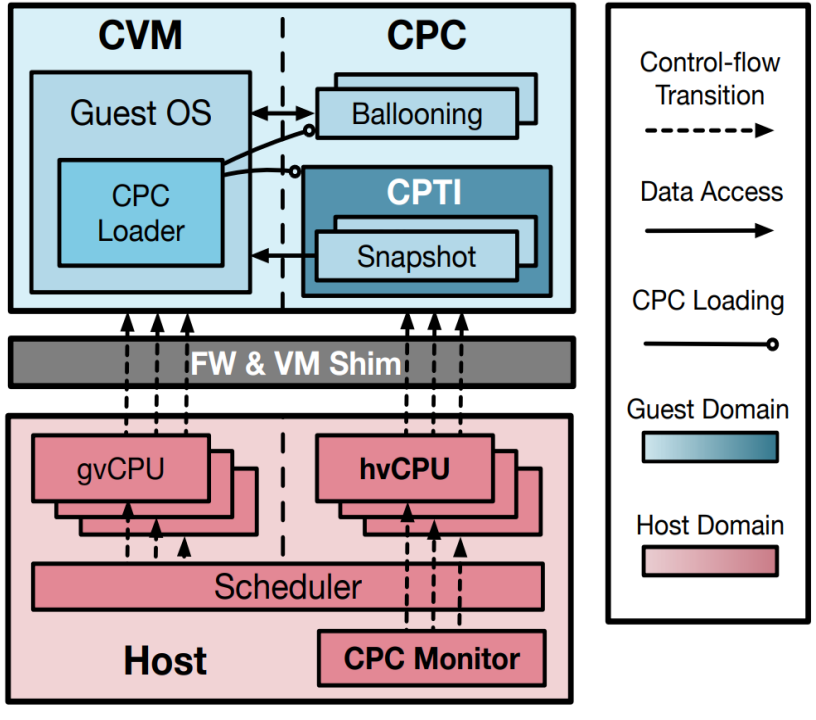
性能测试

- 热迁移场景对比AMD官方方案：
  - AES-GCM算法软件实现，加速达到**55.90倍**
    - 移植mbedtls
  - AESNI加速后，加速达到**69.47倍**
    - 现在大部分开销是GCM编码
  - 要是AES-GCM彻底被加速成普通的内存拷贝呢？
    - 用memcpy代替加密，直接拷贝出明文来模拟
    - 16.41秒，加速比为**123.44倍**
  - VM变多，加速比继续翻倍



总结

- 本文提出机密过程调用
  - Confidential Procedure Calls
  - 将宿主系统**调度vCPU线程**的语义扩展为**调用运维过程**的语义
- 它能帮助CVM构建一套更**灵活**、**安全**、**高效**的运维方案
  - 云厂商与租户自主定制的运维功能
  - 维护了清晰的安全边界和复用了成熟的机制
  - 显著的性能提升
- 它能够兼容当前全部的CVM方案



感谢各位聆听
欢迎交流

微信

邮箱：chenjiahao@sjtu.edu.cn

IPADS
INSTITUTE OF PARALLEL
AND DISTRIBUTED SYSTEMS