



# 系统虚拟化

SSE202/204: 操作系统原理

苏玉鑫

suyx35@mail.sysu.edu.cn

助教：龙玉丹 单诗雯 毛晨希 沈志轩 郑灿峰 胡伟峰



# 版权信息

- 部分内容来自：上海交通大学并行与分布式系统研究所操作系统课件
  - <https://ipads.se.sjtu.edu.cn/courses/os/>
- 其它参考资料：
  - 清华大学操作系统公开课
    - <https://open.163.com/newview/movie/courseintro?newurl=ME1NSA351>
    - 介绍标准内容，适合考研
  - 南京大学计算机软件研究所
    - <http://jyywiki.cn/OS/2025/>
    - <https://space.bilibili.com/202224425/channel/detail?sid=192498>
    - 比较有趣



# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



# 软件连接多方信息

➤ 手机软件



进港						出港					
航班	始发	计划到达	实际到达	航站楼	状态	航班	始发	计划到达	实际到达	航站楼	状态
ZH3242	绵阳南郊	14:30	16:21	T2	起飞 >	MF1275	绵阳南郊	14:30	16:21	T2	起飞 >
CZ8216	盐城南洋	14:35	14:29	T2	到达 >	MF4136	盐城南洋	14:35	14:29	T2	到达 >
CA4317	成都天府	14:35	14:05	T1	到达 >	ZH4317	成都天府	14:35	14:05	T1	到达 >
TV6217	成都天府	14:35	14:05	T1	到达 >	ZH9860	南京禄口	14:35	14:19	T1	到达 >
KY9860	南京禄口	14:35	14:19	T1	到达 >	CA3884	南京禄口	14:35	14:19	T1	到达 >
SC9860	南京禄口	14:35	14:19	T1	到达 >	HO5248	南京禄口	14:35	14:19	T1	到达 >

➤ 机场

➤ 飞机

➤ 航管局

➤ 气象局



# 计算设备集中与分散的变化

## ➤ 大型机时代

- 集中式计算资源，所有人通过网络连接，共享计算资源
- 20世纪70年代，虚拟化技术兴起(!)



## ➤ PC时代

- 分布式计算资源，每个PC用户独占计算资源
- 20世纪80-90年代，虚拟化技术沉寂



## ➤ 云时代

- 集中式计算资源，所有人通过网络连接，共享计算资源
- 21世纪，虚拟化技术再次兴起



# “天上的”云长什么样?



一台服务器的性能  
可以是普通台式机  
的5~10倍



# 微博宕机史：服务器才是明星流量照妖镜



范冰冰 V

人的一生可能会经历各种告别，在我们的相遇中收获的爱和温暖都化为了永恒的力量，感谢一路走来你所有的给予，支持和爱。谢谢未来还会有关心和爱护。  
我们不再是我们，我们依然是我们。

49分钟前

收藏

转发 59755

评论 118200

点赞 389543



不要说外面下雪了 ❄️ 😊 啊？ 分手啦

56分钟前

回复 | 点赞 275



FANGGJX 🎉 ???

57分钟前

回复 | 点赞 230



Alyson\_T1N9 🎉 🐥 🎉 程程序员今天要崩溃了

46分钟前

回复 | 点赞

## 程序员今天要崩溃了！



# 宕机不只影响吃瓜

## 你的B站崩了！突发服务器宕机事故



新华社新媒体

发布时间: 2021-07-15 06:33 | 新华社官方帐号



【解说】7月13日晚间，哔哩哔哩（简称“B站”）突发服务器宕机事故，主站、App 以及小程序都无法使用。“b站崩了”话题一度登上微博热搜榜第一。



1924-2024  
中山大學 世纪华诞  
100th ANNIVERSARY  
SUN YAT-SEN UNIVERSITY



#王者荣耀崩了#怪不得一直转圈.....



王者荣耀 I

25-3-28 21:38 发布于 四川

+关注

亲爱的小伙伴

由于服务器异常，部分玩家出现登录异常、对局无法进入的问题，我们正在紧急处理中。

我们将会在官方渠道同步以上问题的处理情况，请关注后续官方公告，我们对给各位召唤师造成影响感到非常的抱歉，感谢您对《王者荣耀》的支持与理解。

由于王者荣耀类的多人实时对战游戏对手机到服务器之间的时延要求极高（时延超 100ms 就很难受了），如果你是王者荣耀的架构设计师，该如何设计各省玩家之间的匹配算法？如何保证周末晚上的游戏体验？

作答

A map of China where each province is colored according to its network latency. A legend on the left shows six color-coded ranges: red for '超时' (Over Time), orange for '>250ms', yellow for '201ms-250ms', light green for '101ms-200ms', medium green for '51ms-100ms', and dark green for '<=50ms'. Most provinces fall into the 51ms-100ms range.

区域/运营商	最快	最慢	平均
全部节点	上海联通 2ms	辽宁大连移动 61ms	22ms
中国电信	上海电信 3ms	新疆昌吉电信 59ms	22ms
中国联通	上海联通 2ms	西藏拉萨联通 56ms	22ms
中国移动	上海移动 4ms	辽宁大连移动 61ms	22ms
华东地区	上海联通 2ms	福建福州电信 47ms	15ms
华南地区	广东广州电信 6ms	海南海口联通 41ms	20ms
华中地区	河南郑州联通 13ms	湖南长沙移动 25ms	18ms
华北地区	天津联通 2ms	河北承德移动 41ms	12ms
西南地区	贵州贵阳移动 24ms	西藏拉萨联通 56ms	38ms
西北地区	陕西西安联通 17ms	新疆昌吉电信 59ms	31ms
东北地区	辽宁大连联通 17ms	辽宁大连移动 61ms	25ms
港澳台	中国香港 13ms	中国台湾 32ms	22ms



# 优秀的软件服务应有弹性伸缩的能力





# 日常与峰值的矛盾

➤ 假设：

- 1万台服务器可以满足微博网站日常运转
- 应付明星八卦事件需要10万台服务器

➤ 新浪要买10万台服务器维持微博运转么？

- 大瓜又不是天天有
- 浪费资源

➤ 更多日常与峰值的矛盾：

- 双11时的淘宝
- 春运时12306火车票网站

➤ 呼唤云计算：资源共享、时分复用

12306崩了，火车票买不了





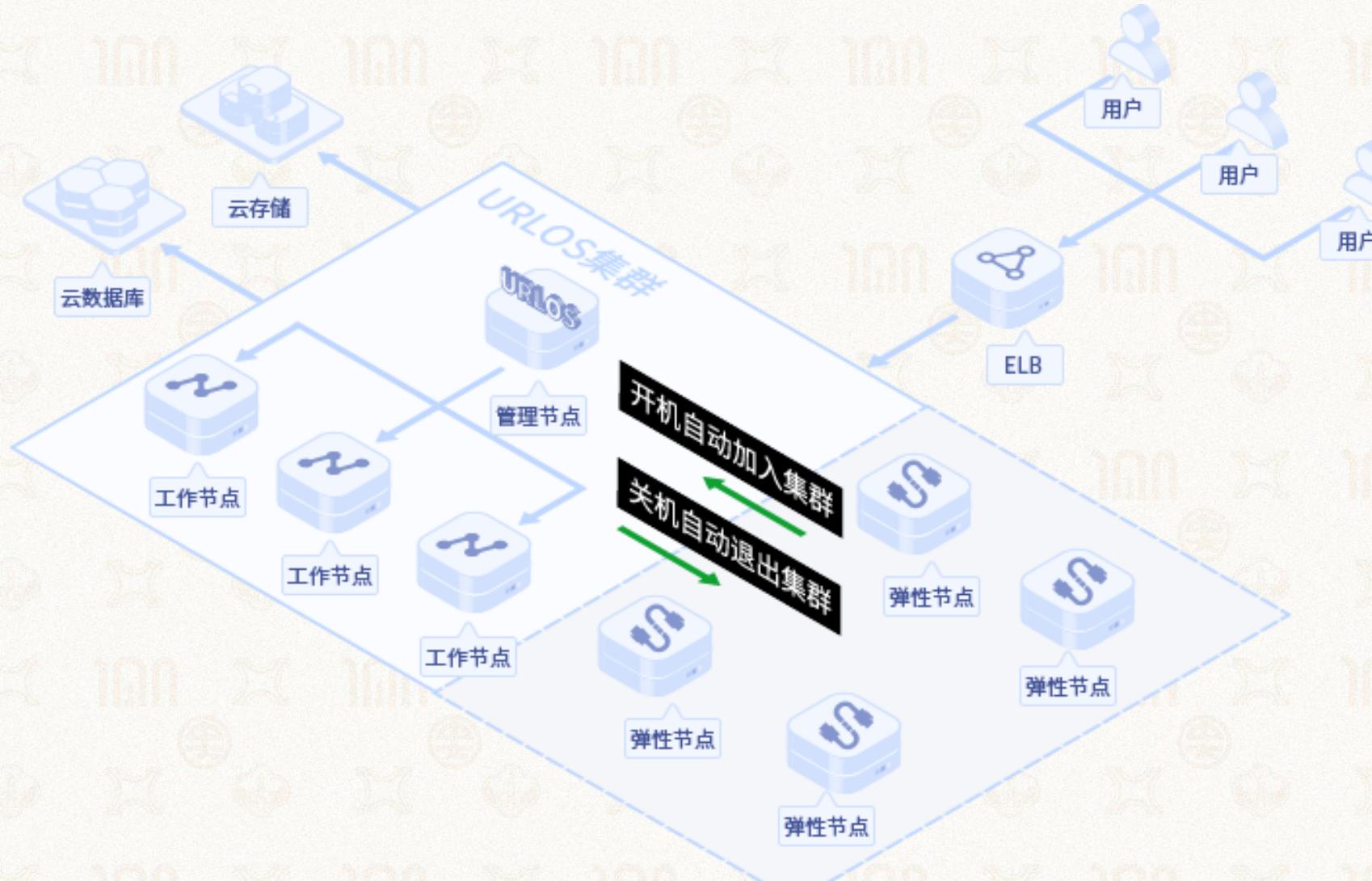
# 现代IT公司的部署方式：云

- 云服务器代替物理服务器
- 云服务器配置与物理服务器一致
- 所有云服务器维护由服务商提供
  
- 对于用户来说：
  - 按需租赁、无需机房租赁费
  - 无需雇佣物理服务器管理人员
  - 可以快速低成本地升级服务器

入门级	企业级
共享型	突发性能型
<b>ECS共享型 s6</b> vCPU 2~8核 100%性能基线，性能强劲，超高性价比，广泛适用于建...  实例 1核2G 地域 华东1 (杭州) 系统盘 40G 高效云盘 带宽 1M 购买时长 3 个月	<b>共享标准型 s6</b> 1vCPU 1G 100%性能基线，性能强劲，超高性价比，广泛适用于建站...  实例 1核1G 地域 华北3 (张家口) 系统盘 40G 高效云盘 带宽 1M 购买时长 1 年
<b>3.6折</b> 限1台 续1年3.6折 ¥104.76 /3 个月 省 ¥291.00 /3 个月  <a href="#">立即购买</a>	<b>766.80 /年</b>  <a href="#">立即购买</a>
<b>共享标准型 s6</b> 1vCPU 2G 100%性能基线，性能强劲，超高性价比，广泛适用于建站...  实例 1核2G 地域 华南1 (深圳) 系统盘 40G 高效云盘 带宽 1M 购买时长 1 年	<b>共享标准型 s6</b> 2vCPU 8G 100%性能基线，性能强劲，超高性价比，广泛适用于建站...  实例 2核4G 地域 华北2 (北京) 系统盘 40G 高效云盘 带宽 1M 购买时长 1 年
  <a href="#">立即购买</a>	  <a href="#">立即购买</a>

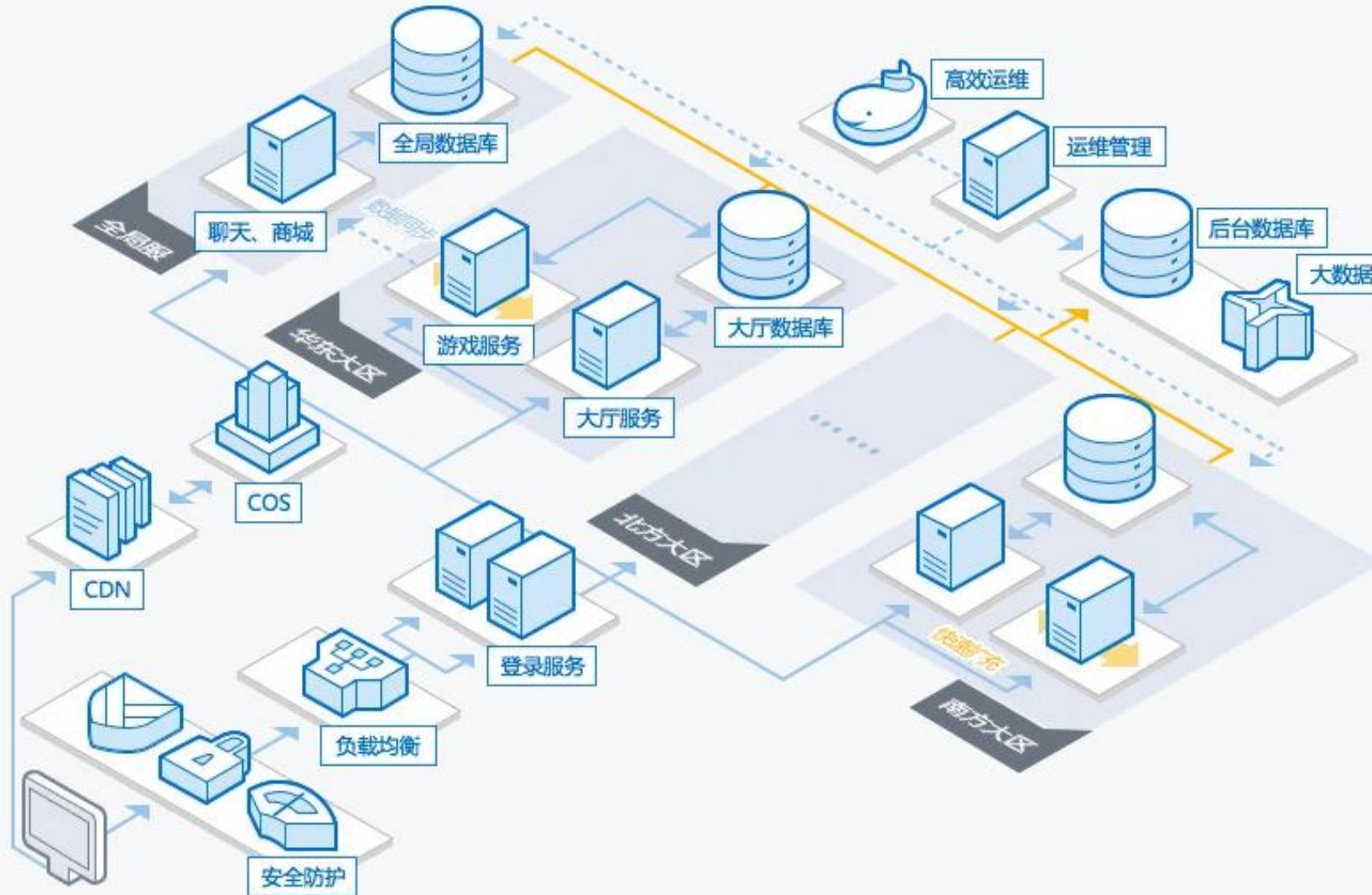


# 弹性伸缩不只是开机、关机这么简单





# 云计算的精髓在于资源共享，时分复用



- 全国是一套大型软件系统
- 软件可以让多个应用共享服务器
  - 每个应用都可以拥有充足的计算资源
- 双11之夜不可能是12306的抢票高峰



# 系统虚拟化是云计算的核心支撑技术

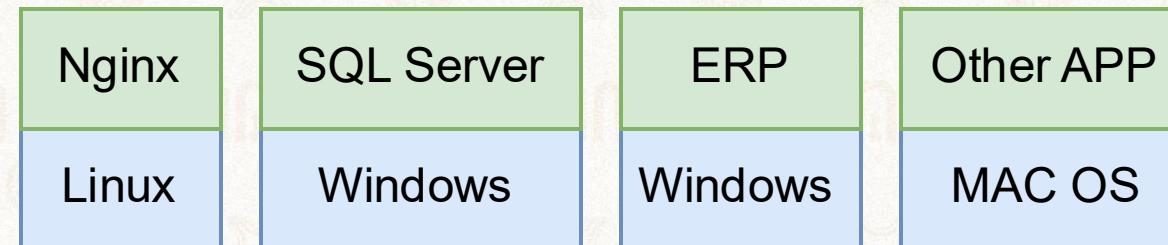
➤ 新引入的一个软件层

➤ 上层是操作系统（虚拟机）

➤ 底层硬件与上层软件解耦

➤ 上层软件可在不同硬件之间切换

- 可自由迁移、快速扩缩



虚拟化层

硬件



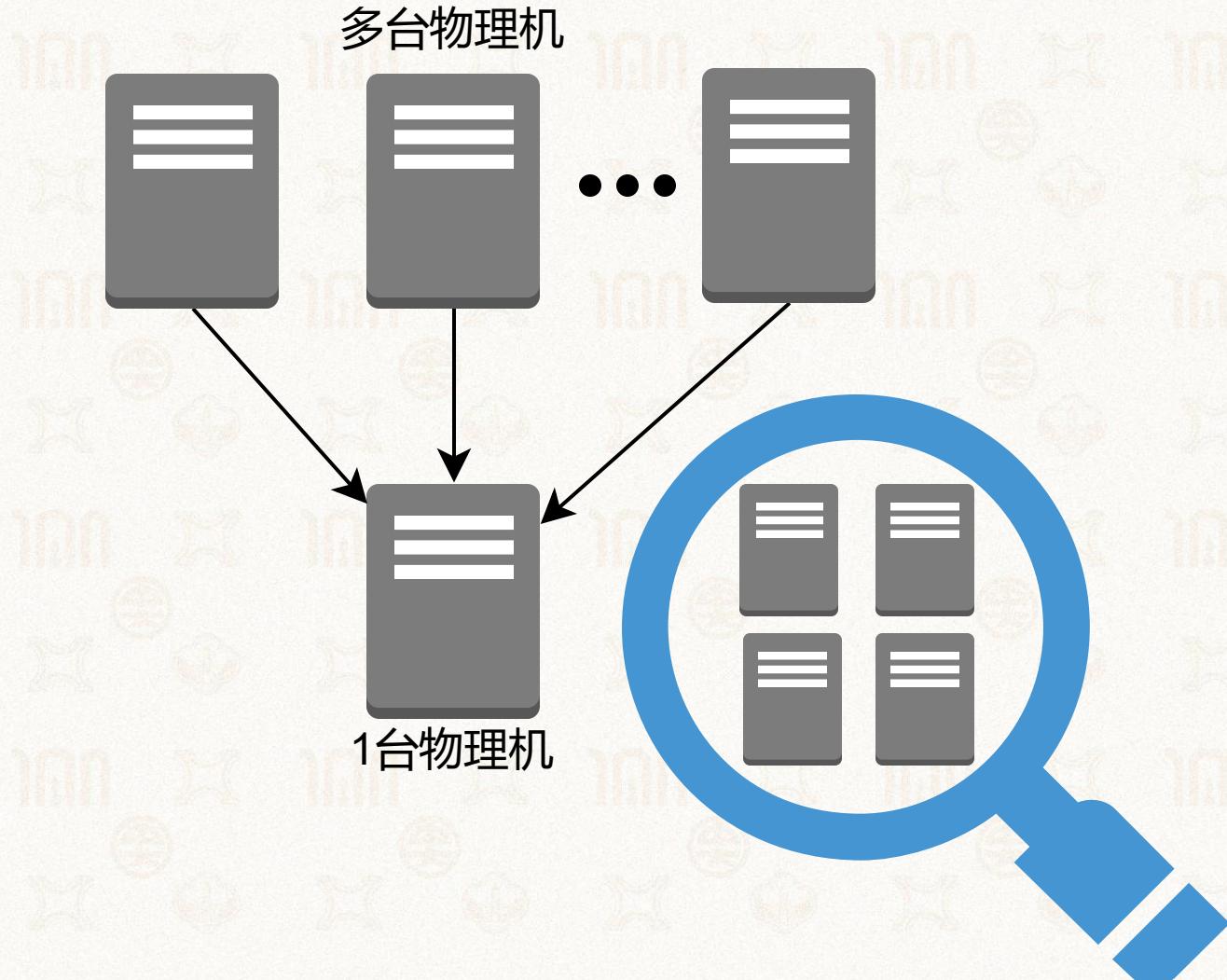
# 大纲

- 虚拟化概述
  - 为什么要用虚拟化
  - 虚拟化的优势
- 什么是系统虚拟化
  - 虚拟机监控器
  - 虚拟化的类型
- CPU虚拟化
  - 下陷
  - 三种软件虚拟化方法
  - 硬件虚拟化



# 虚拟化优势-1：服务器整合

- 单个物理机资源利用率低
  - CPU利用率通常仅20%
- 利用系统虚拟化进行资源整合
  - 一台物理机同时运行多台虚拟机
- 提升物理机资源利用率
- 降低云服务提供商的成本
  - 好的云可以靠“超售”赚钱





## 虚拟化优势-2：方便程序开发



### ➤ 调试操作系统

- 单步调试操作系统
- 查看当前虚拟硬件的状态
  - 寄存器中的值是否正确
  - 内存映射是否正确
- 随时修改虚拟硬件的状态

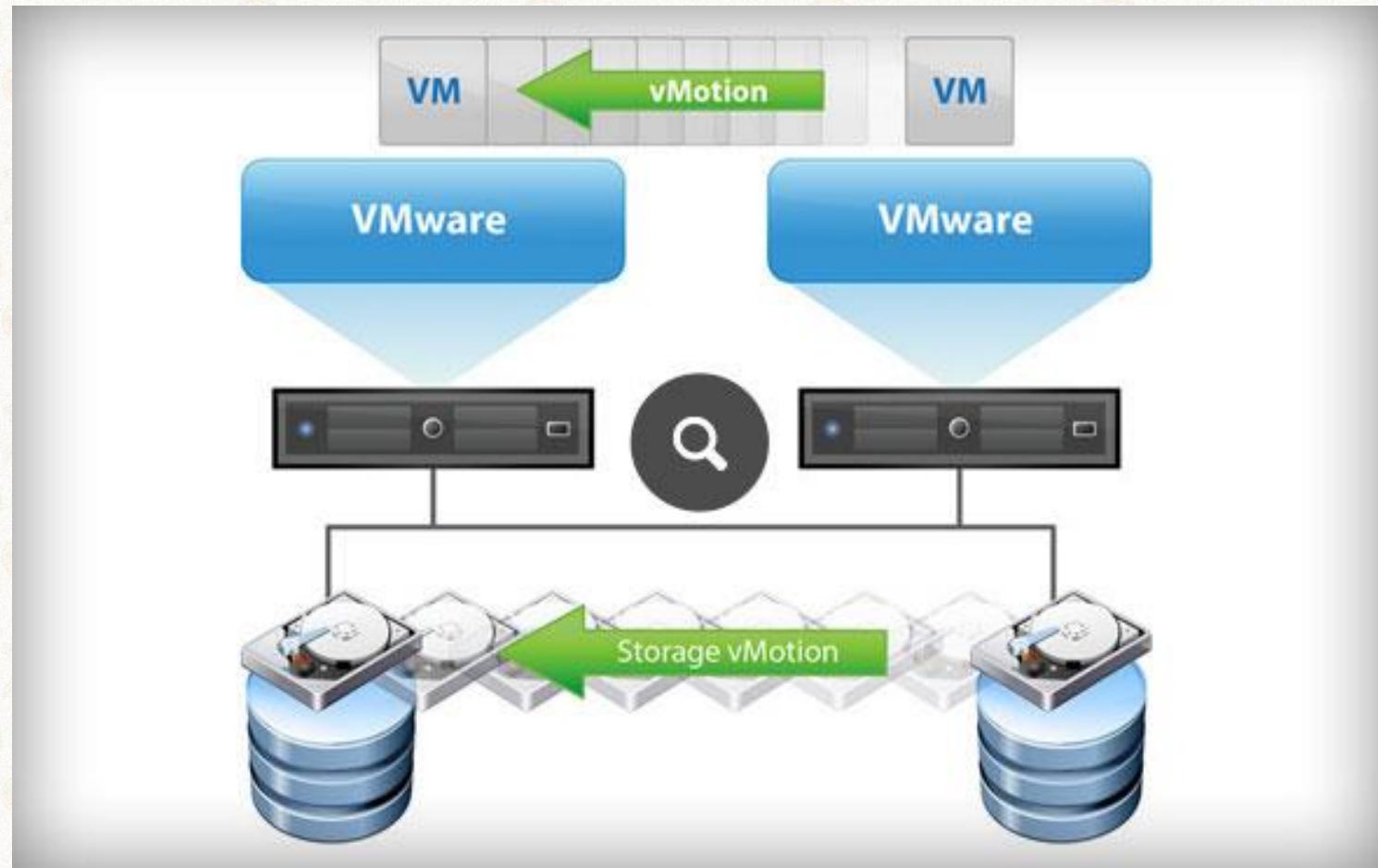
### ➤ 测试应用程序的兼容性

- 可以在一台物理机上同时运行在不同的操作系统
- 测试应用程序在不同操作系统上的兼容性



## 虚拟化优势-3：简化服务器管理

- 通过软件接口管理虚拟机
  - 创建、开机、关机、销毁
  - 方便高效
- 虚拟机热迁移
  - 方便物理机器的维护和升级





# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



# 操作系统中的接口层次: ISA

## ➤ ISA层

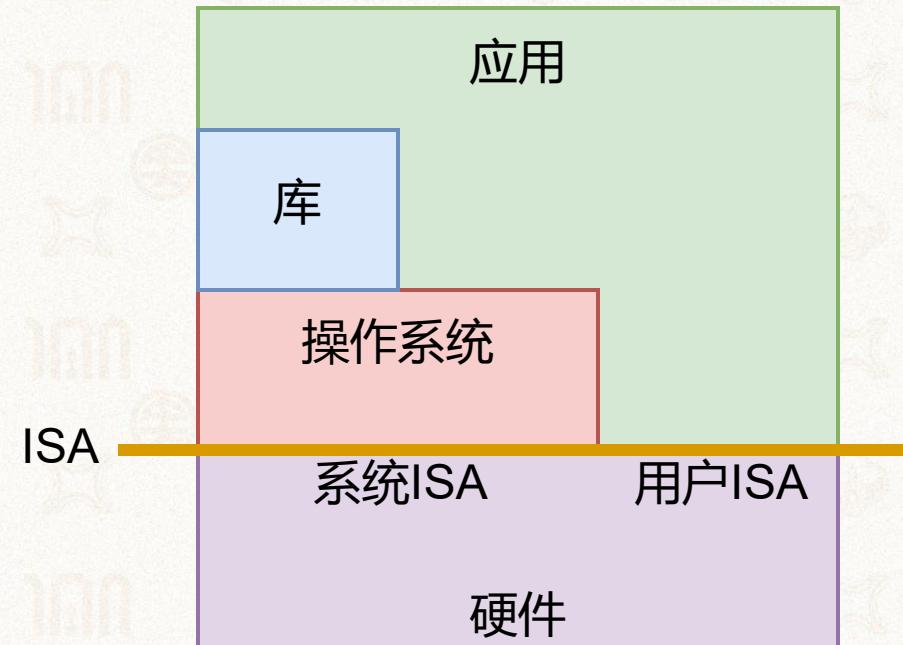
- Instruction Set Architecture
- 区分硬件和软件

## ➤ 用户ISA

- 用户态和内核态程序都可以使用
- `mov x0, sp`
- `add x0, x0, #1`

## ➤ 系统ISA

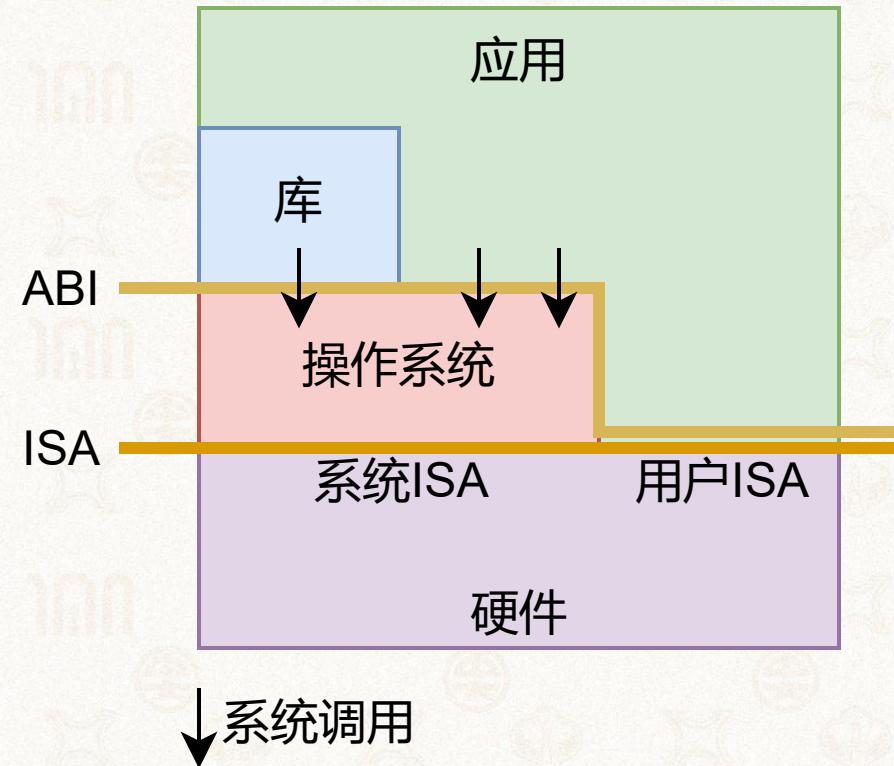
- 只有内核态程序可以使用
- 和特殊寄存器相关
- 和CPU状态相关PSTATE
- `msr vbar_el1, x0`





# 操作系统中的接口层次: ABI

- Application Binary Interface
- 提供操作系统服务或硬件功能
- 包含用户ISA和系统调用





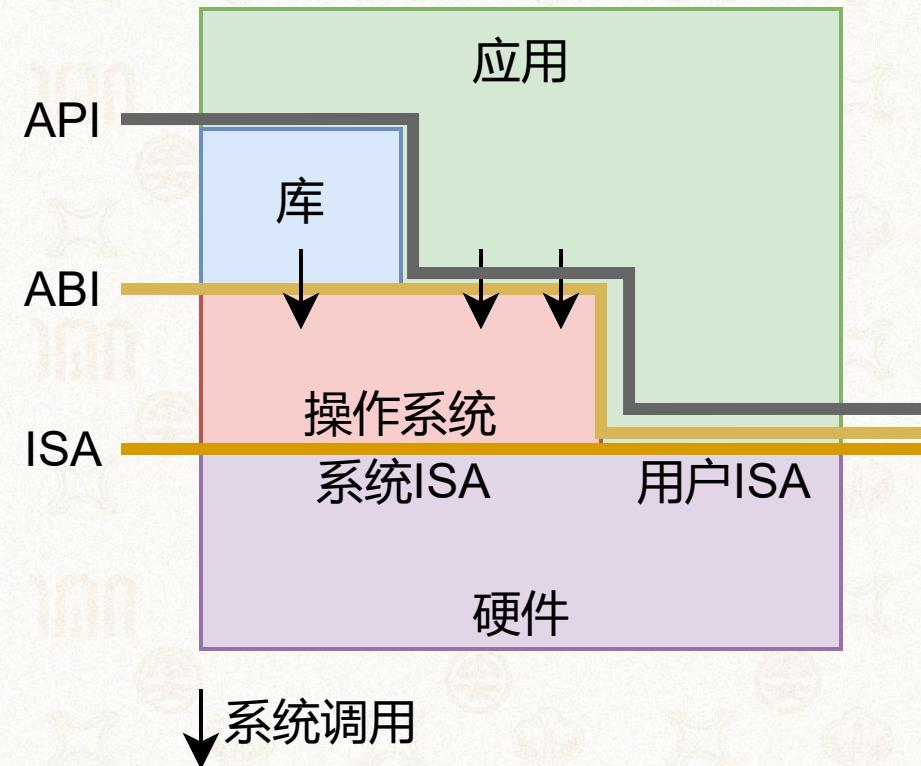
# 操作系统中的接口层次: API

➤ Application Programming Interface

➤ 不同用户态库提供的接口

➤ 包含库的接口和用户ISA

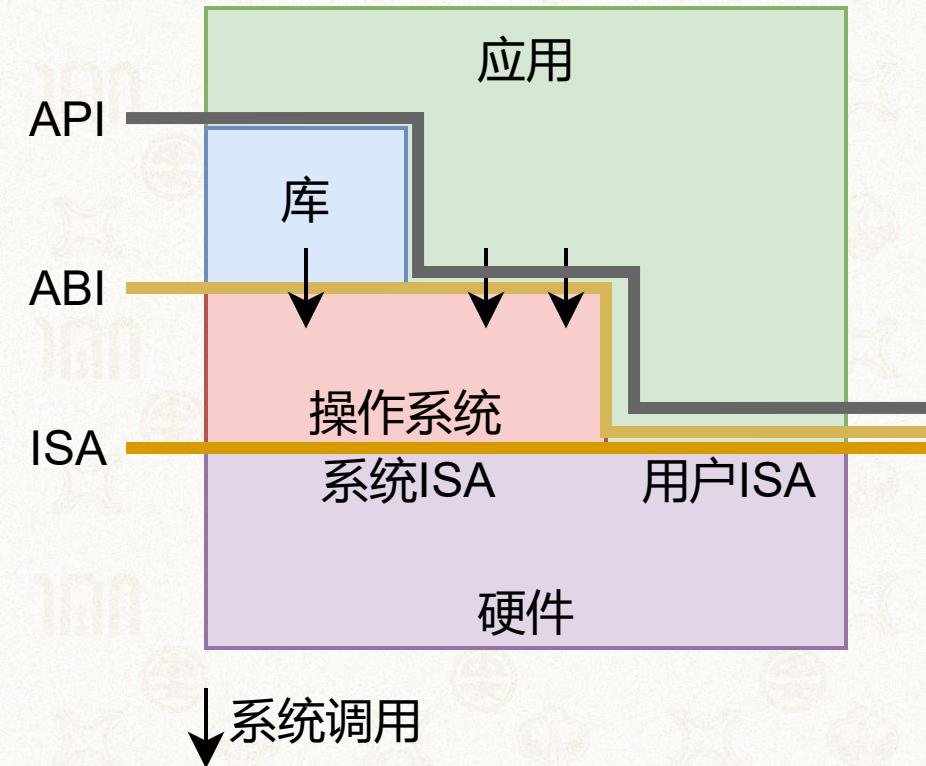
➤ UNIX环境中的c lib:  
• 支持UNIX/C编程语言





# 这些程序用了哪层接口?

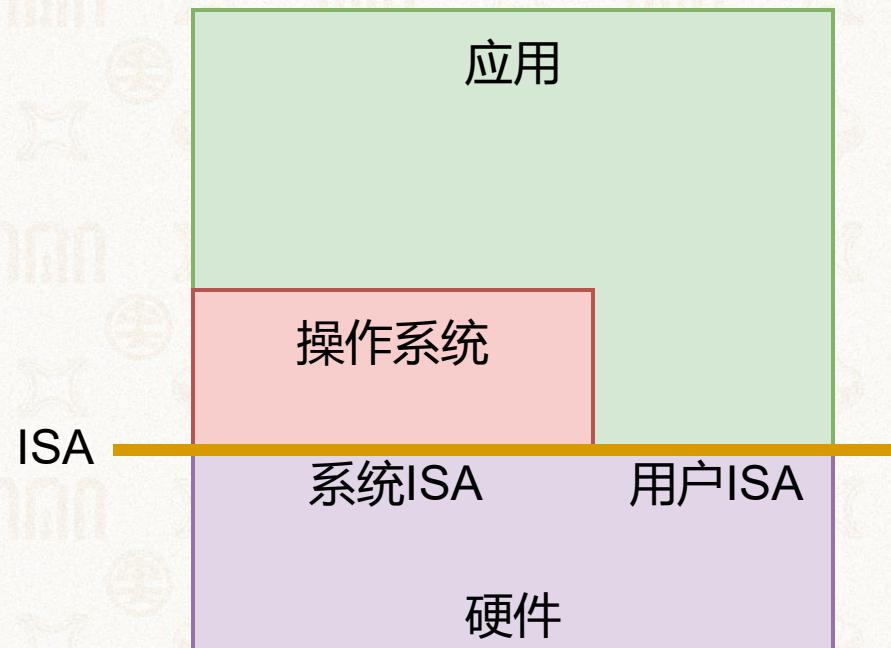
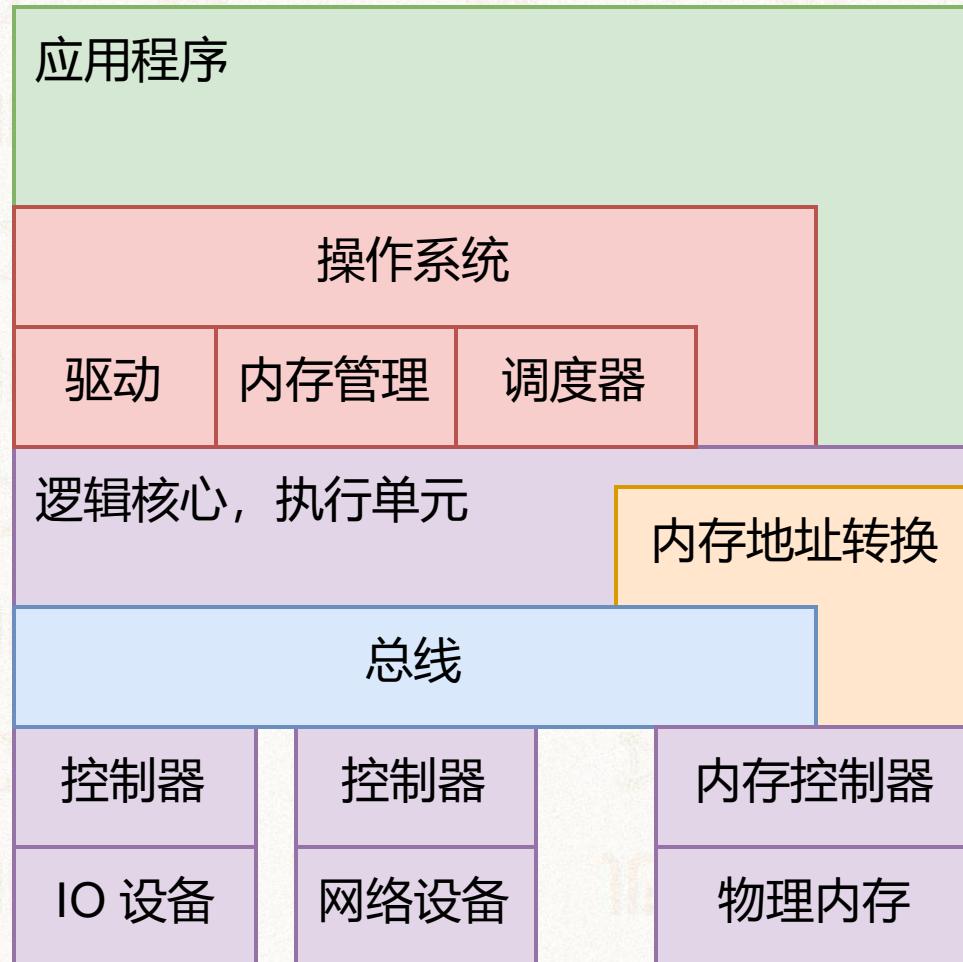
- Hello world
- Web game
- Dota
- Office 2019
- Windows 11
- Java applications
- ChCore





# 如何定义虚拟机(Virtual Machine)?

- 从操作系统角度看"Machine"
- ISA 提供了操作系统和Machine之间的界限





# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

- 设备模拟
- 半虚拟化
- 设备直通

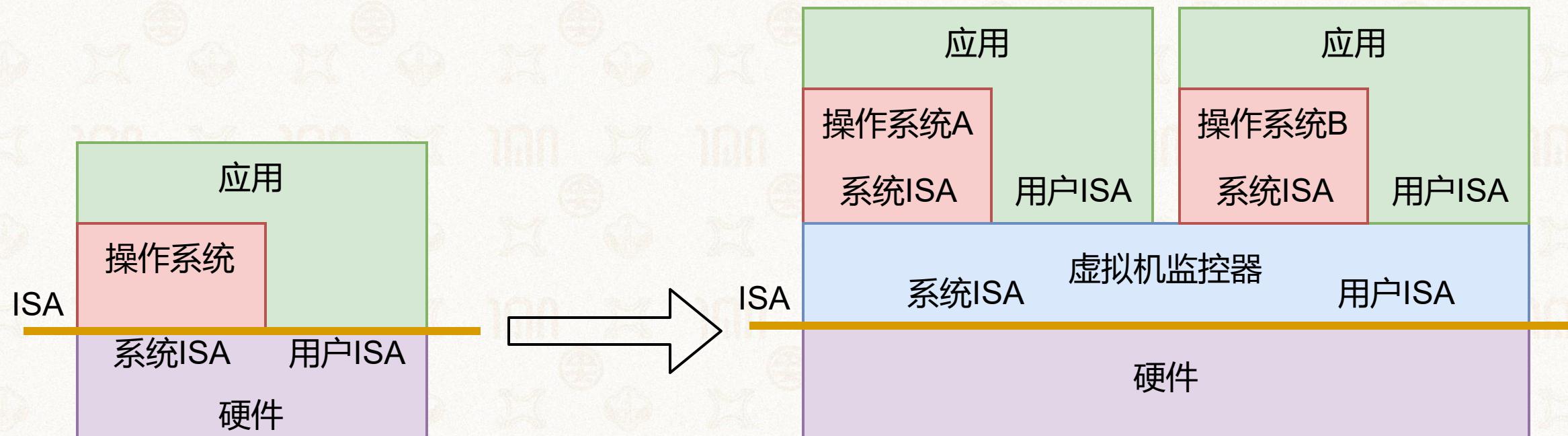
## ➤ 案例：QEMU/KVM



# 虚拟机和虚拟机监控器

## ➤ 虚拟机监控器(Virtual Machine Monitor VMM, Hypervisor)

- 向上层虚拟机暴露其所需要的ISA
- 可同时运行多台虚拟机(VM)

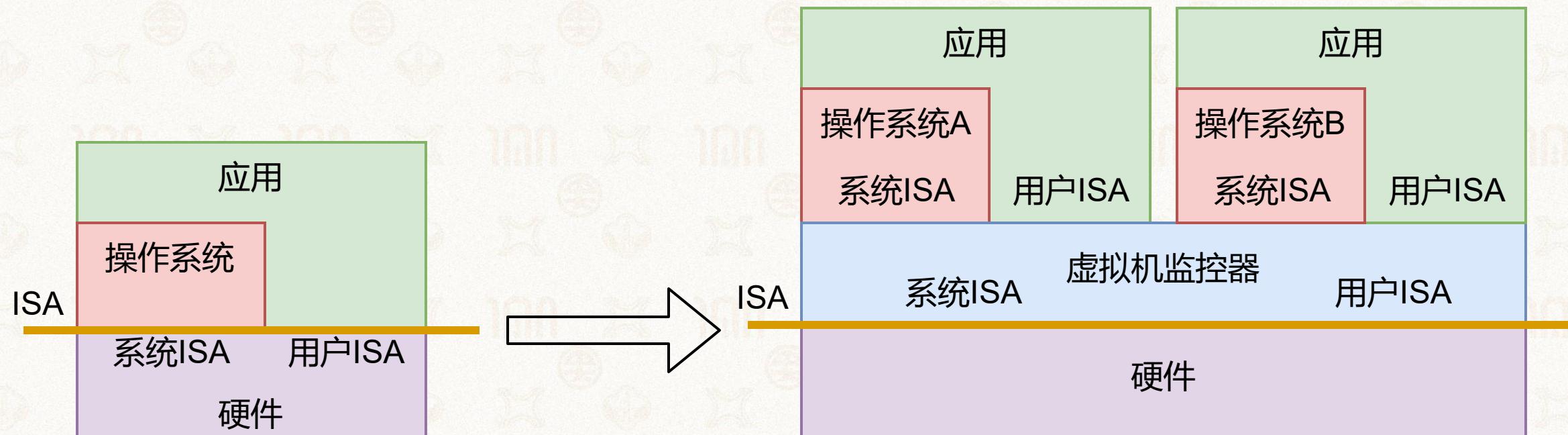




# 系统虚拟化的标准

## ➤ 高效 系统虚拟化的三个特性

- 为虚拟机内程序提供与该程序原先执行的硬件完全一样的接口
- 虚拟机只比在无虚拟化的情况下性能略差一点
- 虚拟机监控器控制所有物理资源





# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- **虚拟化的类型**

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



# 虚拟机监控器的分类

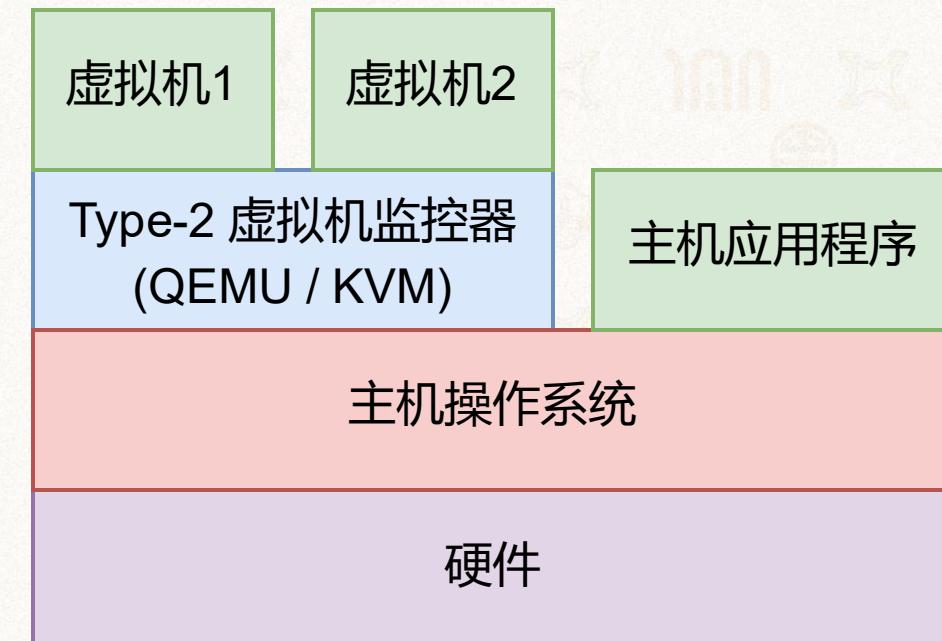
## ➤ Type-1

- 直接运行在硬件之上
  - 充当操作系统的角色
  - 直接管理所有物理资源
  - 实现调度、内存管理、驱动等功能
- 性能损失较少
- 例如Xen, VMware ESX Server



## ➤ Type-2

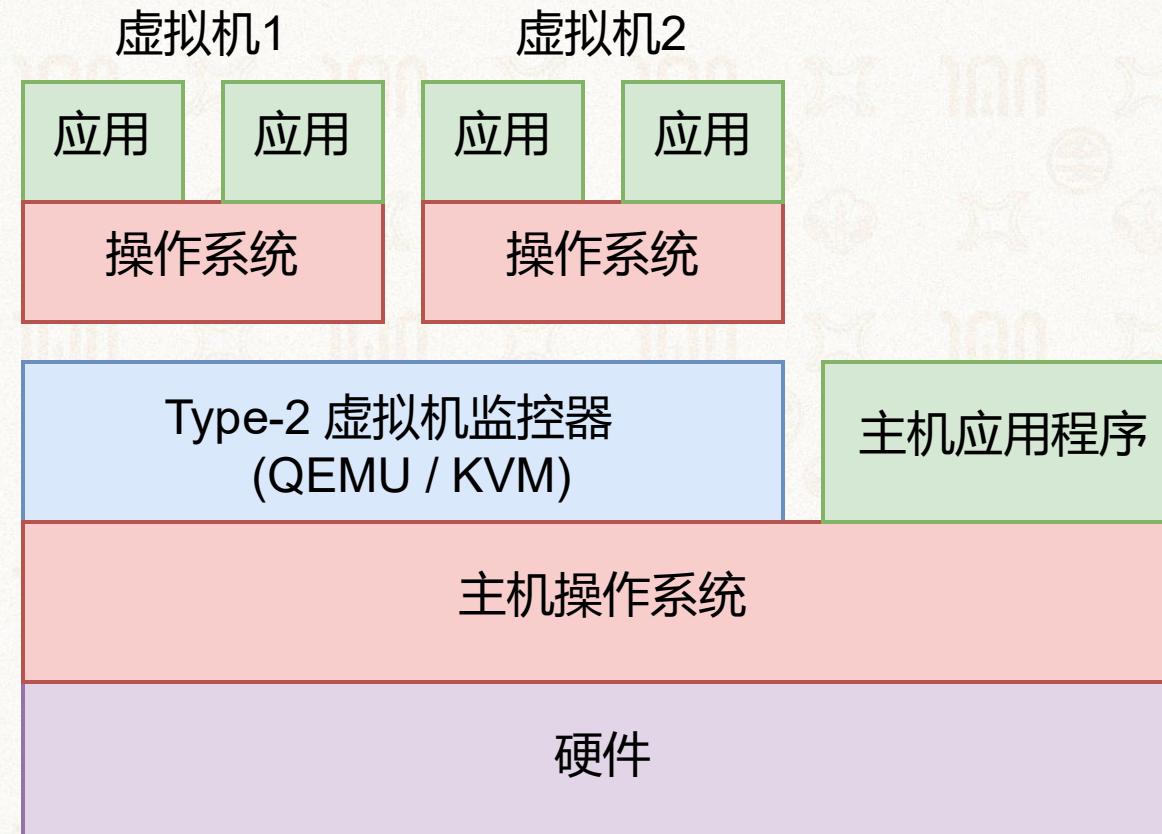
- 依托于主机操作系统
  - 主机操作系统管理物理资源
  - 虚拟机监控器以进程/内核模块的形态运行
- 易于实现和安装
- 例如QEMU/KVM, VMware/VirtualBox





# Type-2的优势

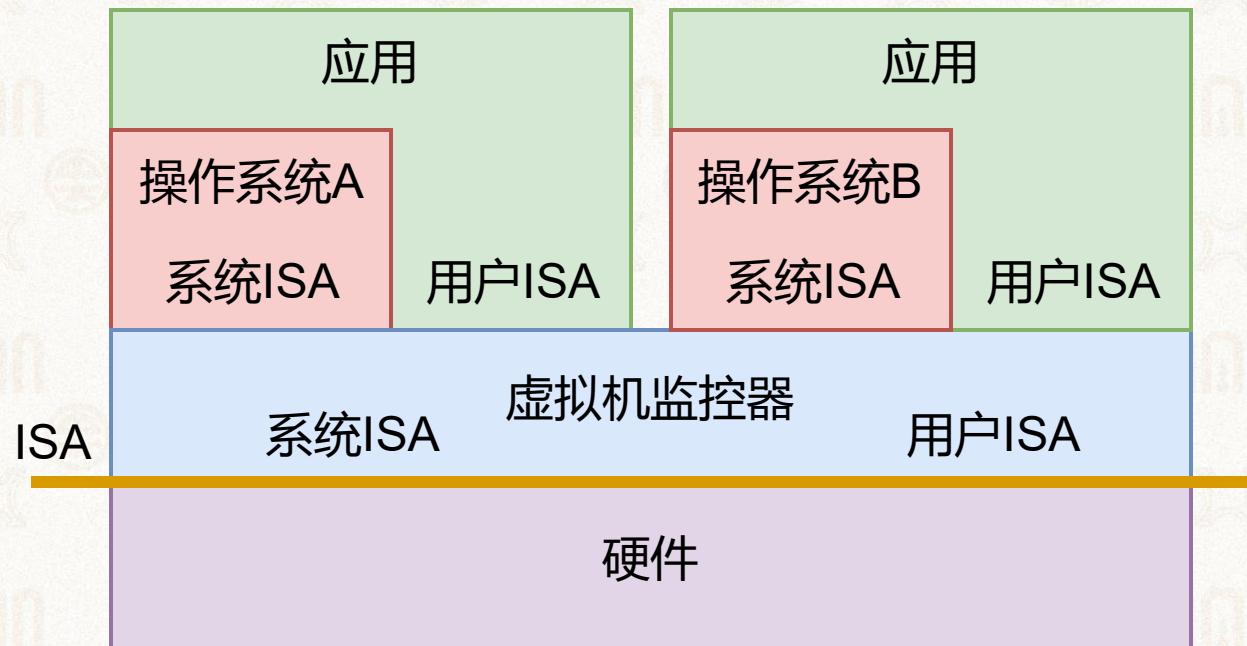
- 在已有的操作系统之上将虚拟机当做应用运行
- 复用主机操作系统的大部分功能
  - 文件系统
  - 驱动程序
  - 处理器调度
  - 物理内存管理





# 如何实现系统虚拟化?

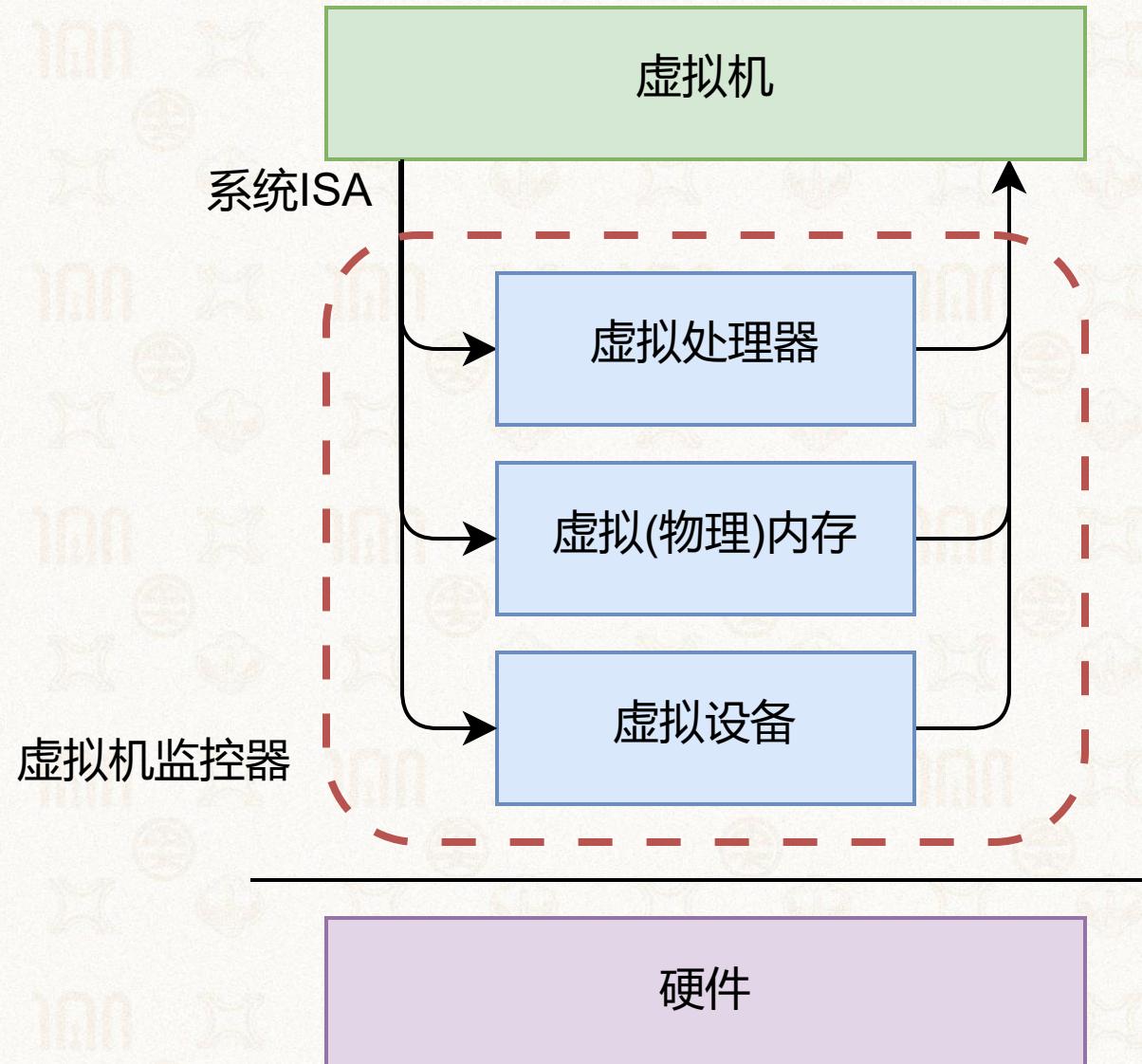
- 难点在于管好系统ISA
  - 多是临界资源
- 系统ISA:
- 读写敏感寄存器
  - sctrl\_el1、ttbr0\_el1/ttbr1\_el1...
- 控制处理器行为
  - 例如: WFI(陷入低功耗状态)
- 控制虚拟/物理内存
  - 打开、配置、安装页表
- 控制外设
  - DMA、中断





# 系统虚拟化的流程

- 第一步
  - 捕捉所有系统ISA并下陷(Trap)
- 第二步
  - 由具体指令实现相应虚拟化
    - 控制虚拟处理器行为
    - 控制虚拟内存行为
    - 控制虚拟设备行为
- 第三步
  - 回到虚拟机继续执行





# 系统虚拟化技术

## ➤ 处理器虚拟化

- 捕捉系统ISA
- 控制虚拟处理器的行为

## ➤ 内存虚拟化

- 提供“假”物理内存的抽象

## ➤ 设备虚拟化

- 提供虚拟的I/O设备



# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

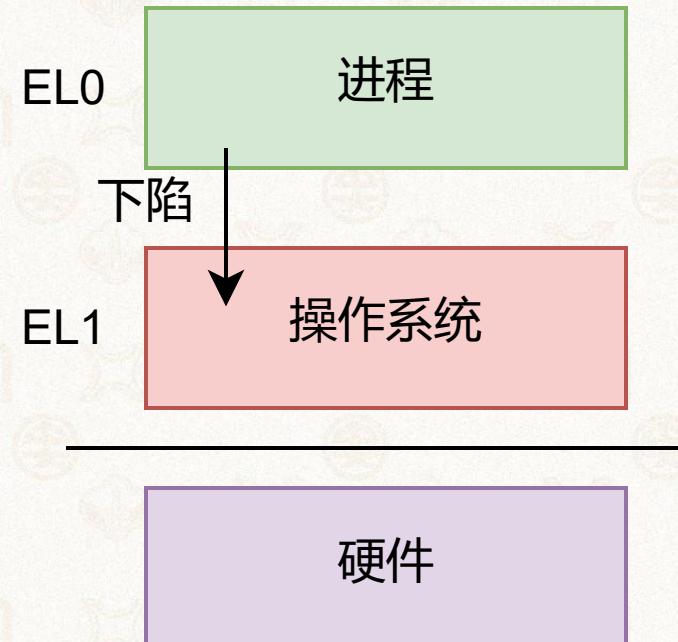
- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



# 回顾：ARM的特权级

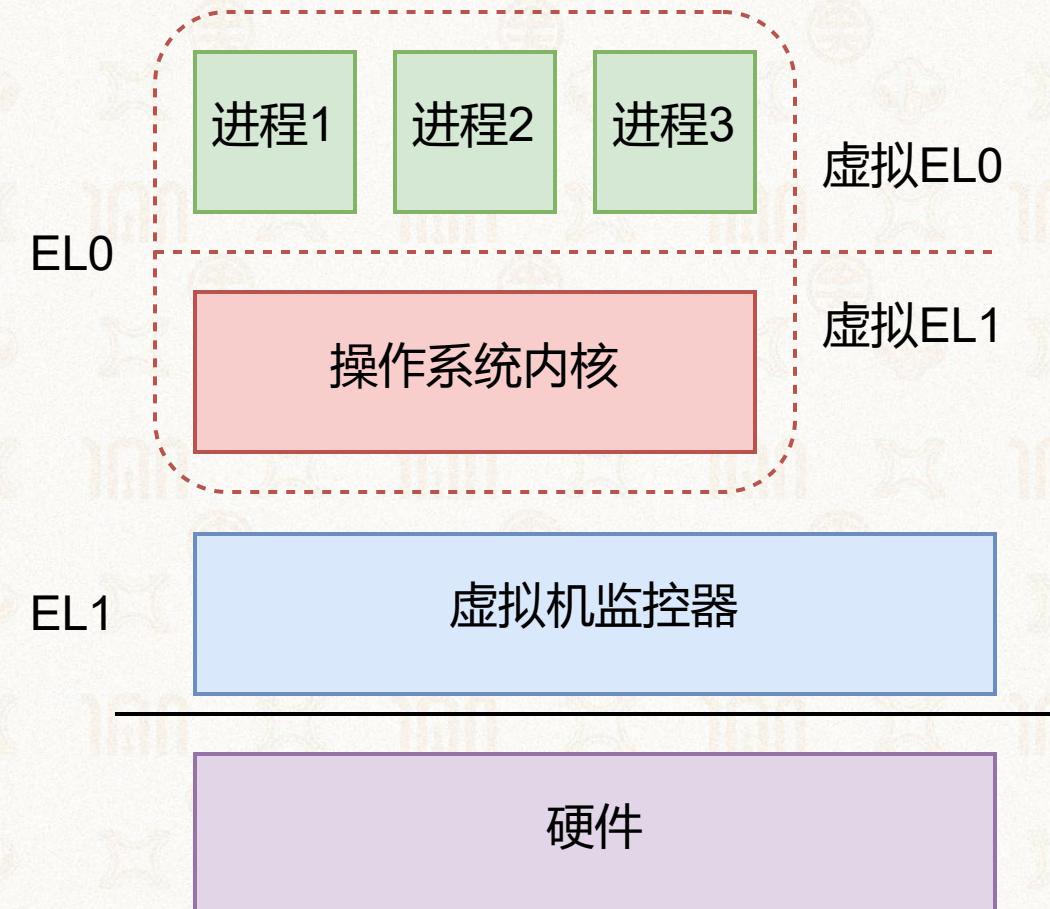
- EL0: 用户态进程
- EL1: 操作系统内核
- 处理器何时会从EL0进入到EL1?
  - 执行系统调用，具体指令为特权调用：`svc (supervisor call)`
  - 应用执行的指令触发了异常
  - CPU收到了外设发来的中断信息





# 处理器虚拟化：一种直接的实现方法

- 将虚拟机监控器运行在EL1
- 将VM操作系统和其上的进程都运行在EL0
  - VM操作系统不知道自己在用户态
- 当操作系统执行系统ISA指令时下陷
  - 写入TTBR0\_EL1
  - 执行WFI指令
  - ...





# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

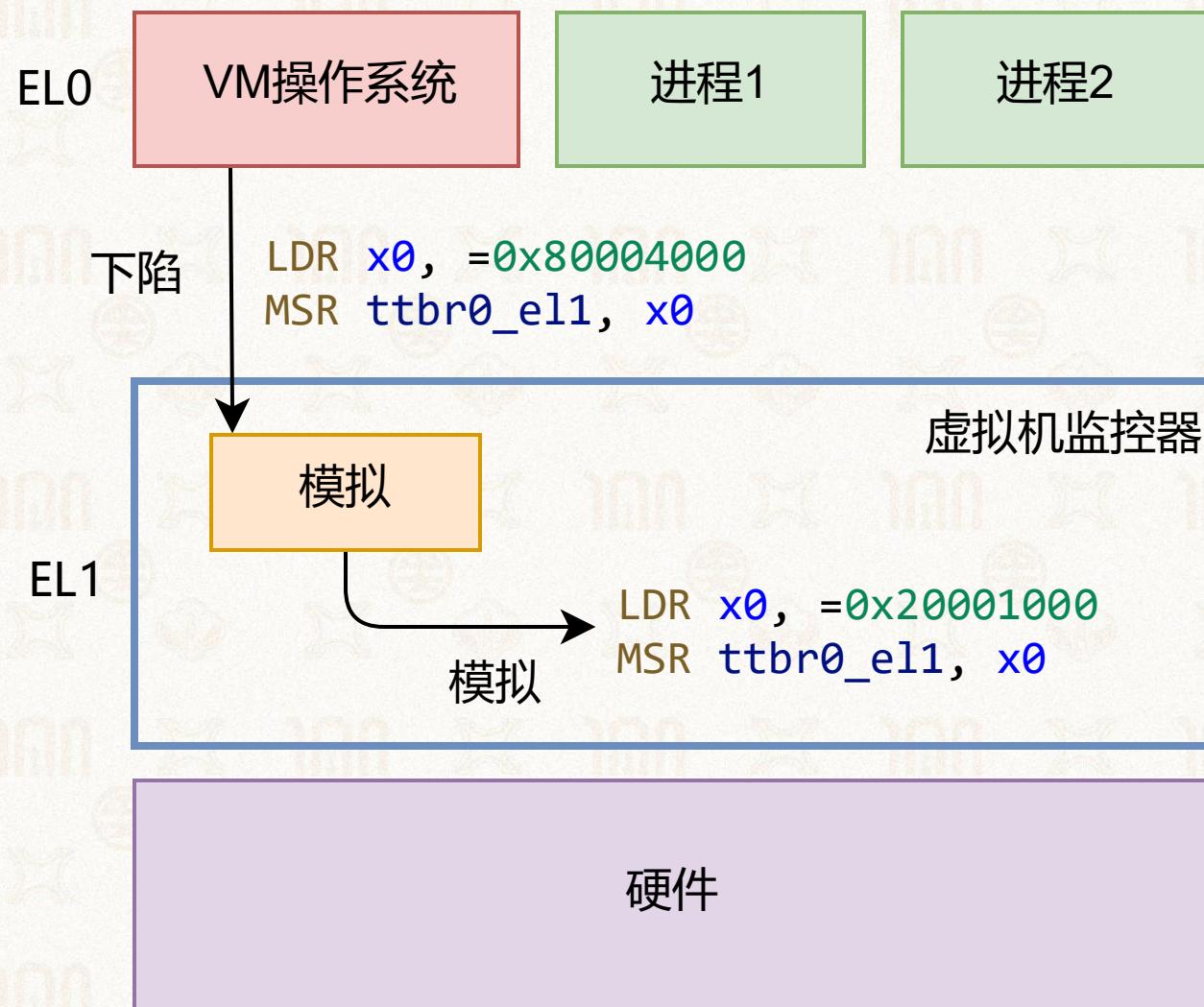
- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



# 下陷(trap)和模拟(emulate)

- 下陷：在用户态EL0执行特权指令将陷入EL1的虚拟机监控器中
- 模拟：这些指令的功能都由虚拟机监控器内的函数安全地实现
- 如果做到以上两点，虚拟化就成功了
  - 也称作可虚拟化架构





## ➤ 敏感指令

- 读写特殊寄存器或更改处理器状态
- 读写敏感内存：例如访问未映射内存、写入只读内存
- I/O指令

## ➤ 特权指令

- 在用户态执行会触发**异常**，并陷入内核态

## ➤ 当所有敏感指令都是特权指令时，才叫作可虚拟化架构

- 可惜ARM不是

当所有的敏感指令在非特权级执行时都会触发下陷时，该CPU架构称作可虚拟化架构。

- A 是
- B 否

提交



# ARM不是严格的可虚拟化架构

- 例子: CPSID/CPSIE指令
- CPSID和CPSIE分别可以关闭和打开中断
- 内核态执行: PSTATE.{A, I, F} 可以被CPS指令修改
- 在用户态执行: CPS 被当做NOP指令, 不产生任何效果
- 不是特权指令
- 类似这样的指令还有很多



# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



# 如何处理这些不会下陷的敏感指令？



- 处理这些不会下陷的敏感指令，使得虚拟机中的操作系统能够运行在用户态（EL0）
- 方法1：解释执行
- 方法2：二进制翻译
- 方法3：半虚拟化
- 方法4：硬件虚拟化（改硬件）



# 方法1：解释执行

➤ 使用软件方法一条条对虚拟机代码进行模拟

- 不区分敏感指令还是其他指令
- 没有虚拟机指令直接在硬件上执行

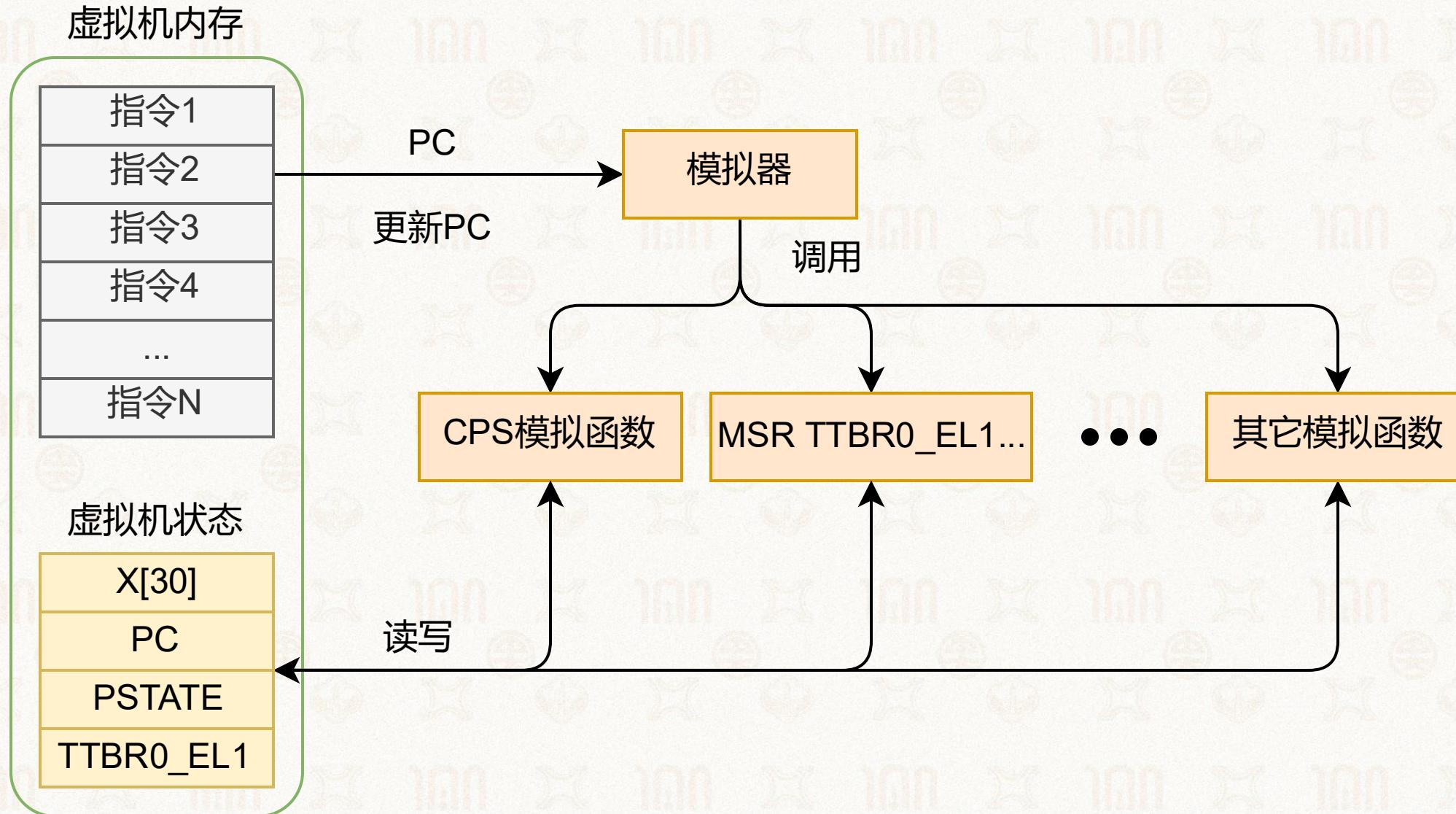
➤ 使用内存维护虚拟机状态

- 例如：使用uint64\_t x[30]数组保存所有通用寄存器的值

```
typedef struct CPUARMState {  
    /* Regs for current mode. */  
    uint32_t regs[16];  
    /* Regs for A64 mode. */  
    uint64_t xregs[32];  
    uint64_t pc;  
    uint32_t pstate;  
    uint32_t aarch64;  
    CPUARMTBFlags hflags;  
    uint32_t uncached_cpsr;  
    uint32_t spsr;  
    /* Banked registers. */  
    uint64_t banked_spsr[8];  
    uint32_t banked_r13[8];  
    uint32_t banked_r14[8];  
    /* These hold r8-r12. */  
    uint32_t usr_regs[5];  
    uint32_t fiq_regs[5];  
    ...
```



# 方法1：解释执行





# 方法1：解释执行

## ➤ 优点：

- 解决了敏感函数不下陷的问题
- 可以模拟不同ISA的虚拟机
- 易于实现、复杂度低

## ➤ 缺点：

- 非常慢：任何一条虚拟机指令都会转换成多条模拟指令

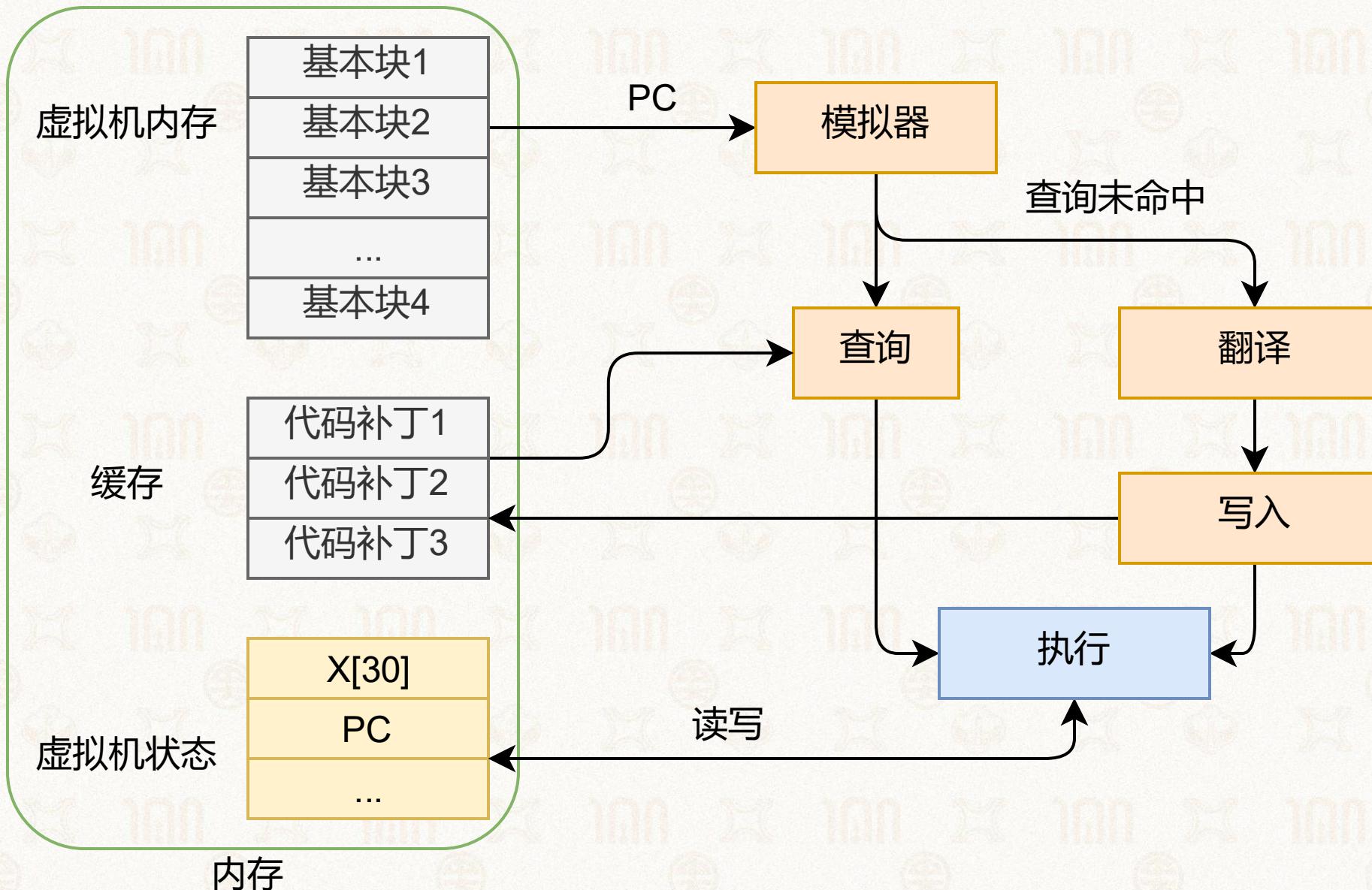


## 方法2：二进制翻译

- 提出两个加速技术
  - 在执行前批量翻译虚拟机指令
  - 缓存已翻译完成的指令
- 使用基本块(Basic Block)的翻译粒度 (为什么?)
  - 每一个基本块被翻译完后叫代码补丁



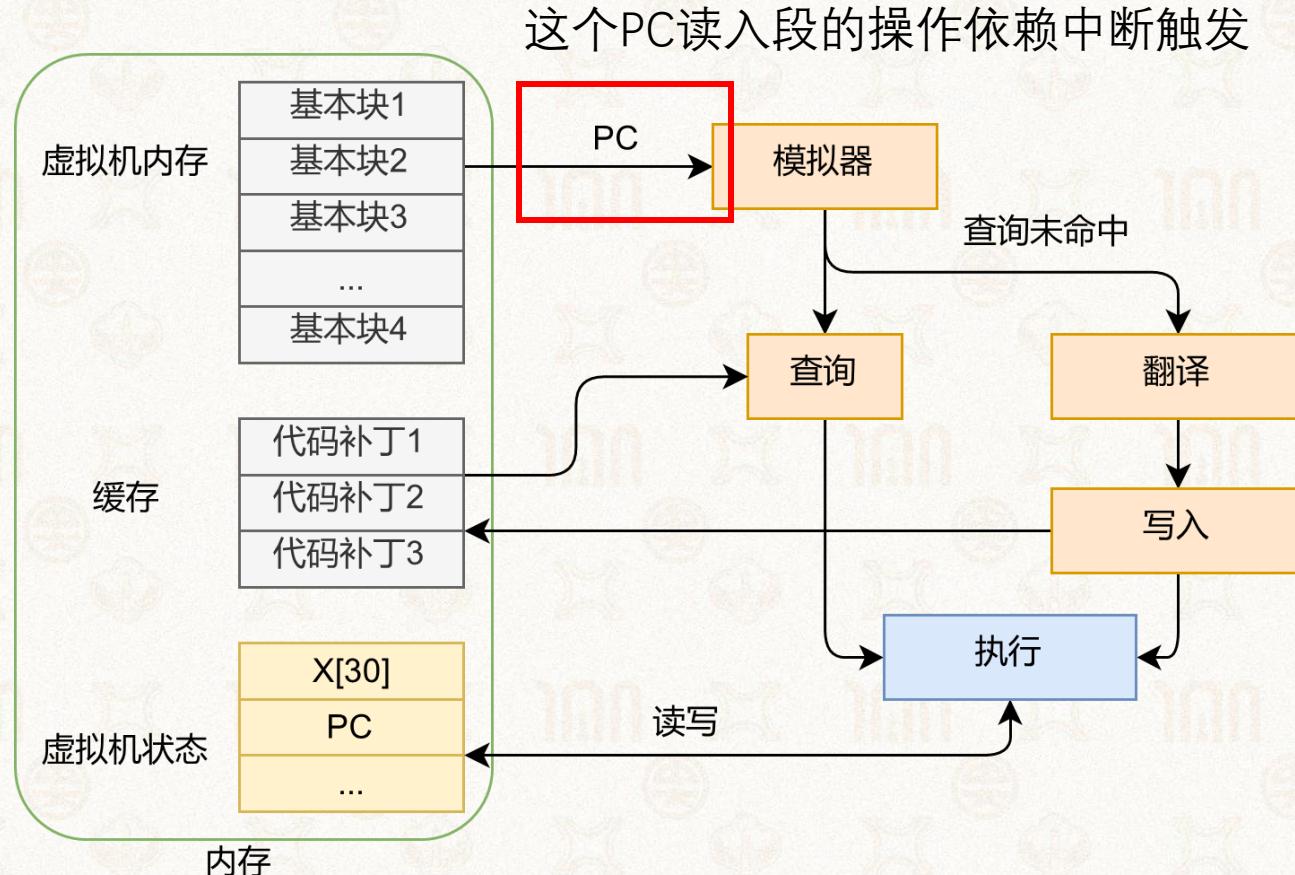
## 方法2：二进制翻译





# 二进制翻译的缺点

- 不能处理自修改的代码(Self-modifying Code)
- 中断插入粒度变大
  - 模拟执行可以在任意指令位置插入虚拟中断
  - 二进制翻译时只能在基本块边界插入虚拟中断





## 方法3：半虚拟化

- 协同设计
  - 让VMM提供接口给虚拟机，称为Hypervisor
  - 修改操作系统源码，让其主动调用VMM接口
- Hypervisor可以理解为VMM提供的系统调用
  - 在ARM中是HVC指令
- 将所有不引起下陷的敏感指令替换成超级调用



# 半虚拟化方法的优缺点

## ➤ 优点：

- 解决了敏感函数不下陷的问题
- 协同设计的思想可以提升某些场景下的系统性能
  - I/O等场景

## ➤ 缺点：

- 需要修改操作系统代码，难以用于闭源系统
  - 比如Windows
- 即使是开源系统，也难以同时在不同版本中实现



# 大纲

## ➤ 虚拟化概述

- 为什么要用虚拟化
- 虚拟化的优势

## ➤ 什么是系统虚拟化

- 虚拟机监控器
- 虚拟化的类型

## ➤ CPU虚拟化

- 下陷
- 三种软件虚拟化方法
- 硬件虚拟化

## ➤ 内存虚拟化

- 影子页表
- 直接页表
- 硬件虚拟化

## ➤ I/O 虚拟化

- 设备模拟
- 半虚拟化
- 设备直通

## ➤ 案例：QEMU/KVM



1924-2024  
中山大學 世纪华诞  
100th ANNIVERSARY  
SUN YAT-SEN UNIVERSITY

1924-2024

# 谢谢

---

微信: suyuxin

钉钉: 苏玉鑫

B站: <https://space.bilibili.com/502854403>

软工集市课程专区: <https://ssemarket.cn/new/course>

匿名提问箱: <https://suask.me/ask-teacher/106/苏玉鑫>