FTF2016
TECHNOLOGY FORUM CHINA

基于ARM的QorIQ SoC上的KVM可铸就卓越的虚拟化性能

FTF-DES-N1887

DIANA CRĂCIUN
软件工程师
2016年5月16日

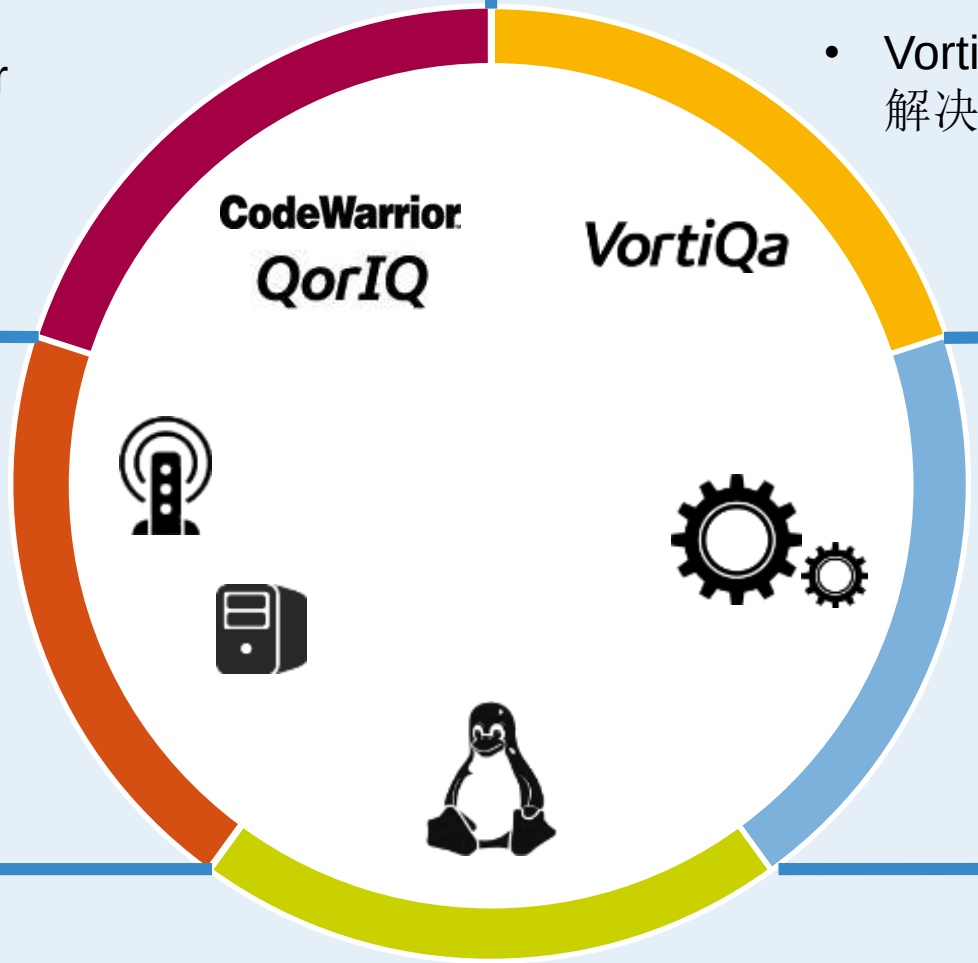
软件产品和服务

开发工具

- CodeWarrior

运行时产品

- VortiQa软件解决方案



解决方案参考

- 物联网网关
- OpenWRT+

集成服务

- 安全咨询
- 强化Linux

Linux®服务

- 商业支持
- 性能调谐

*加快*客户产品上市时间

*提供*商用软件、支持、服务和解决方案

*简化*与恩智浦的软件合作

*创造*成功！

请在线访问：www.nxp.com/networking-services

引领网络中的64位ARM浪潮

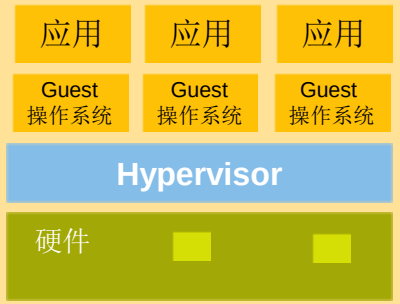
议程

- 简介
- KVM/QEMU
- ARM虚拟化扩展
- ARM上的KVM
- 结果
- 结论

简介

虚拟化和Hypervisor

- **虚拟化** – 能够提供一个抽象层以在单个计算机系统上运行多个操作系统的硬件和软件技术
- **Hypervisor** 是一种可创建和管理能够运行Guest操作系统的虚拟机的软件组件

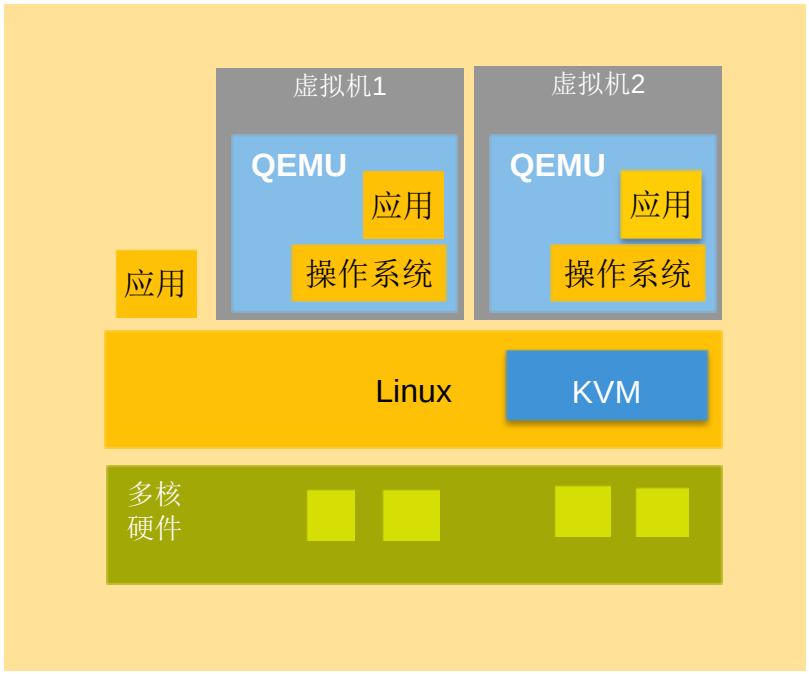


- Hypervisor可运行“裸机”

- Hypervisor集成在主机操作系统中
  - 复用操作系统基础设施
- 主机操作系统可运行其他应用程序

KVM/QEMU

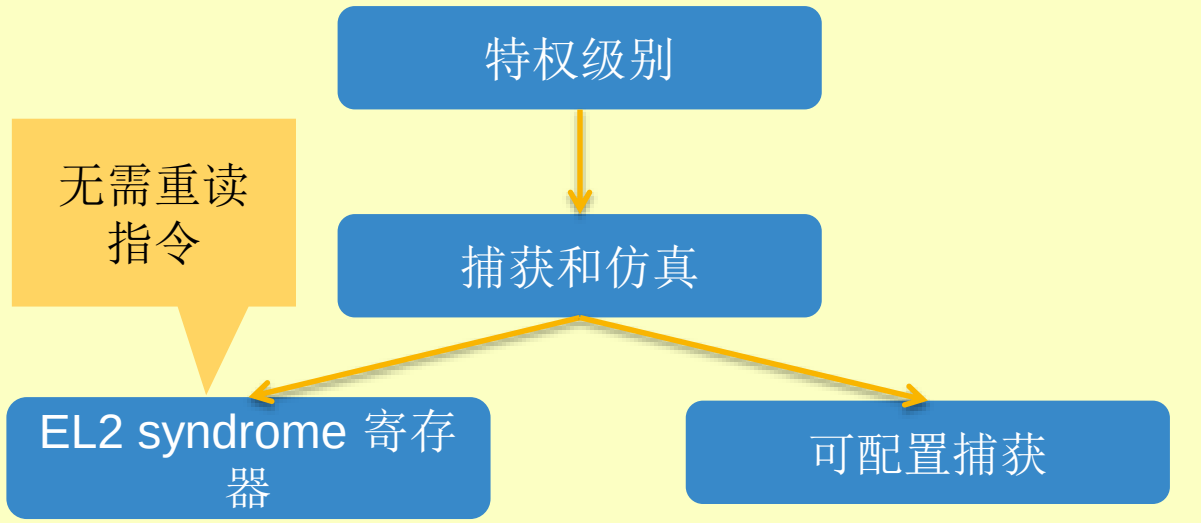
KVM/QEMU

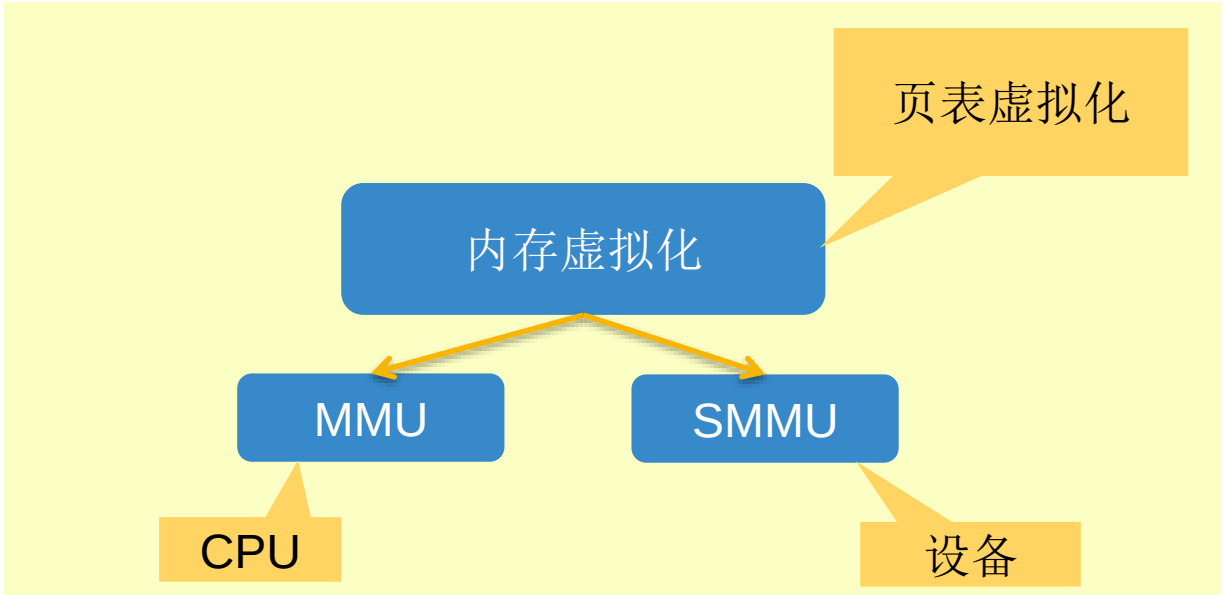


- KVM/QEMU – 基于Linux®内核的开源虚拟化技术
- KVM是一个Linux内核模块
- QEMU是一个使用KVM进行加速的用户空间仿真器
- 同时运行虚拟机和Linux应用程序
- 操作系统无需更改或仅需小幅更改
- 虚拟I/O功能
- Direct / pass through I/O – 为VM分配I/O设备

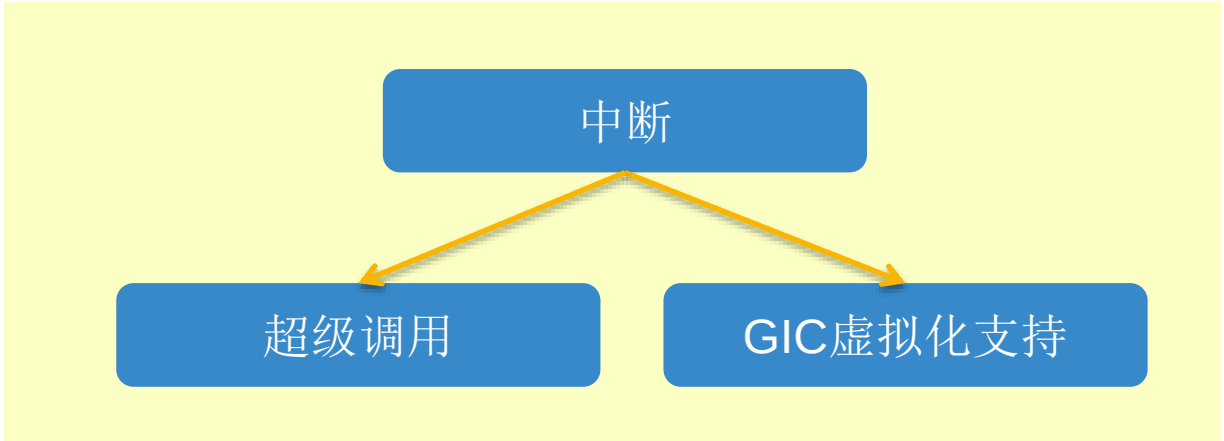
ARM虚拟化扩展

ARM虚拟化扩展

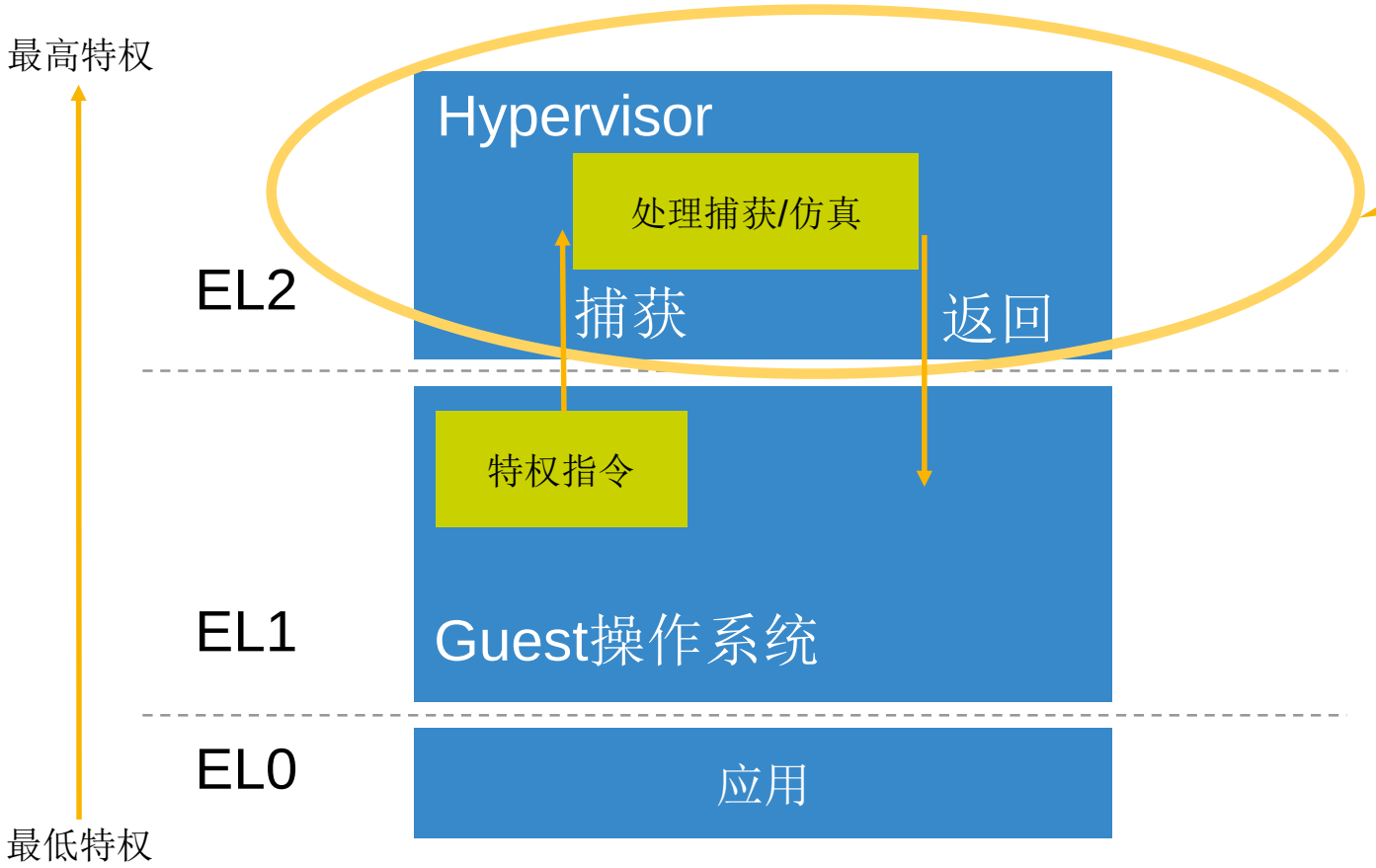

特权级别
无需重读指令
捕获和仿真
EL2 syndrome 寄存器
可配置捕获


页表虚拟化
内存虚拟化
MMU
SMMU
CPU
设备


Guest接口
虚拟定时器


中断
超级调用
GIC虚拟化支持

特权级别

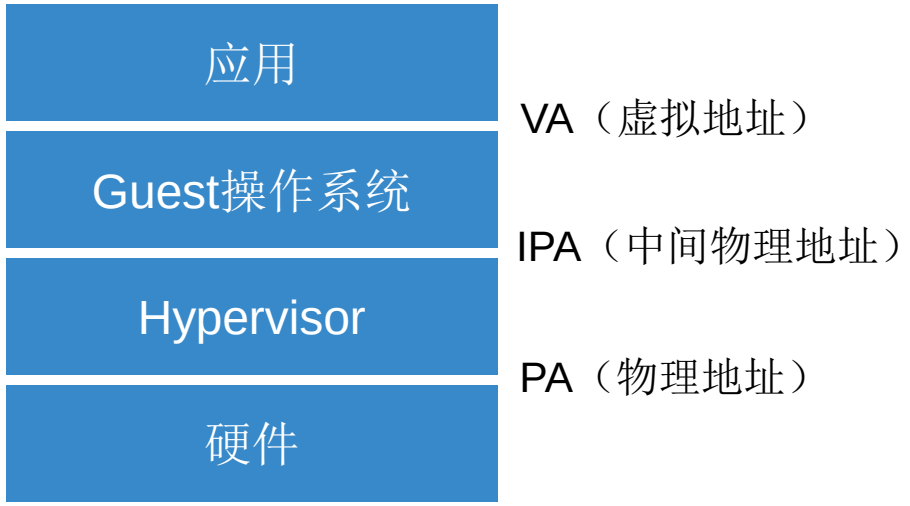


通过异常向EL2过渡

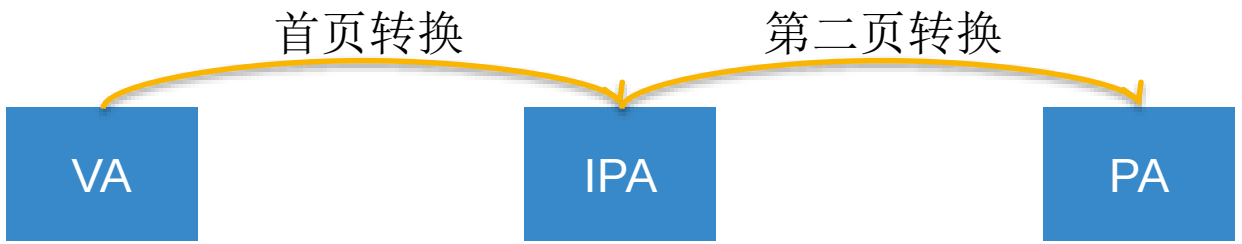
- 捕获和仿真、中断
- 超级调用

为每个EL使用单独的堆栈指针

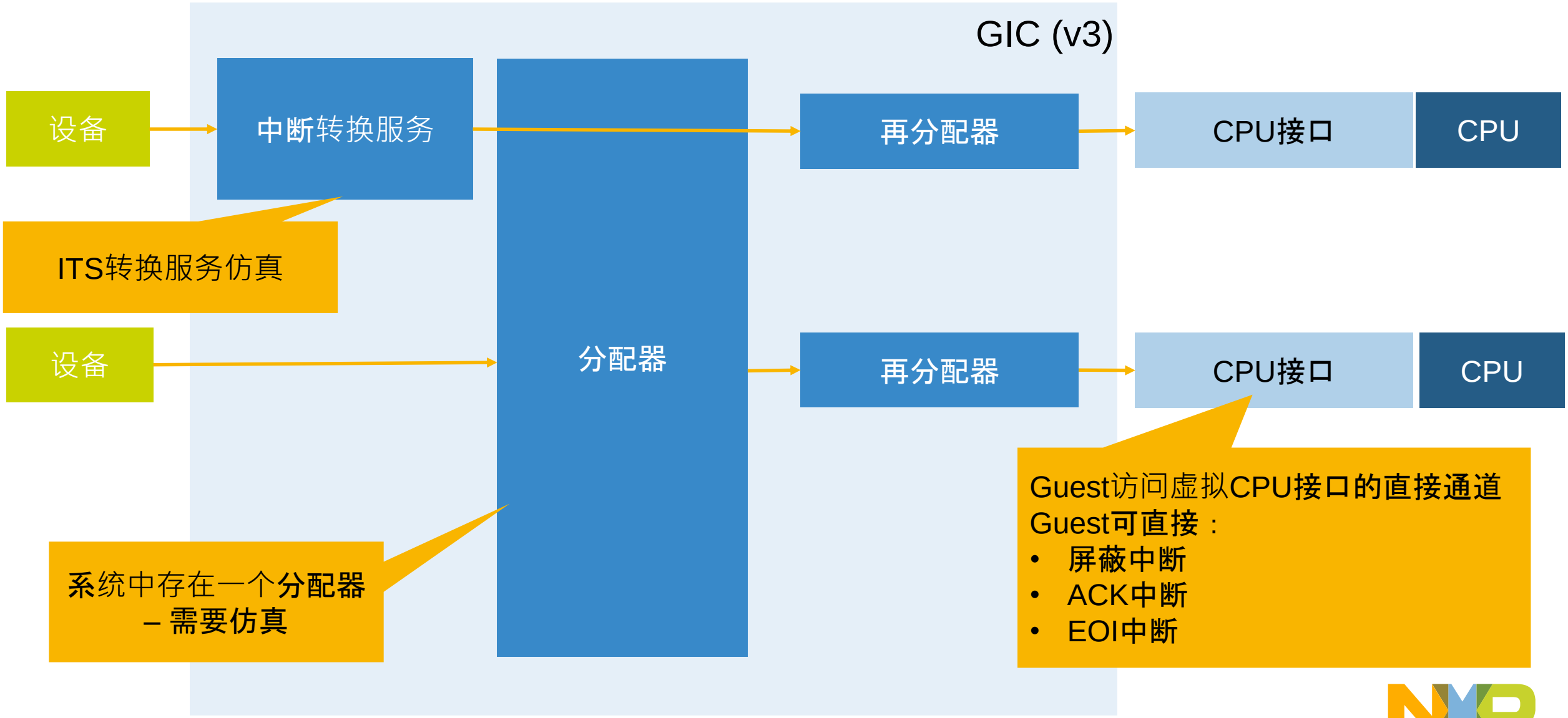
内存虚拟化



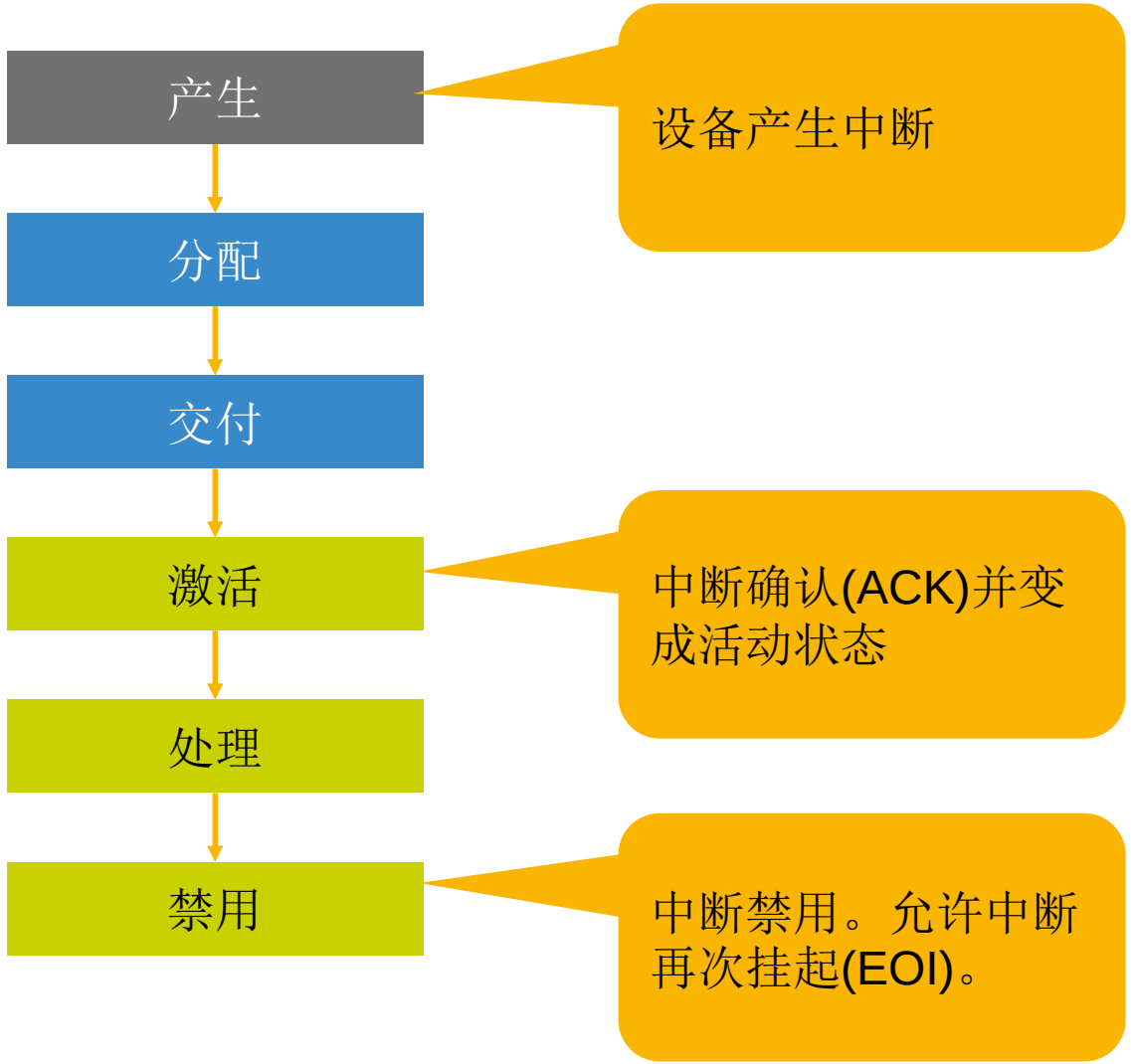
- 2级内存转换
- 首次转换将内存从VA转换至IPA
  - 由Guest所有
- 二级转换从IPA转换至PA
  - 表项由Hypervisor维护

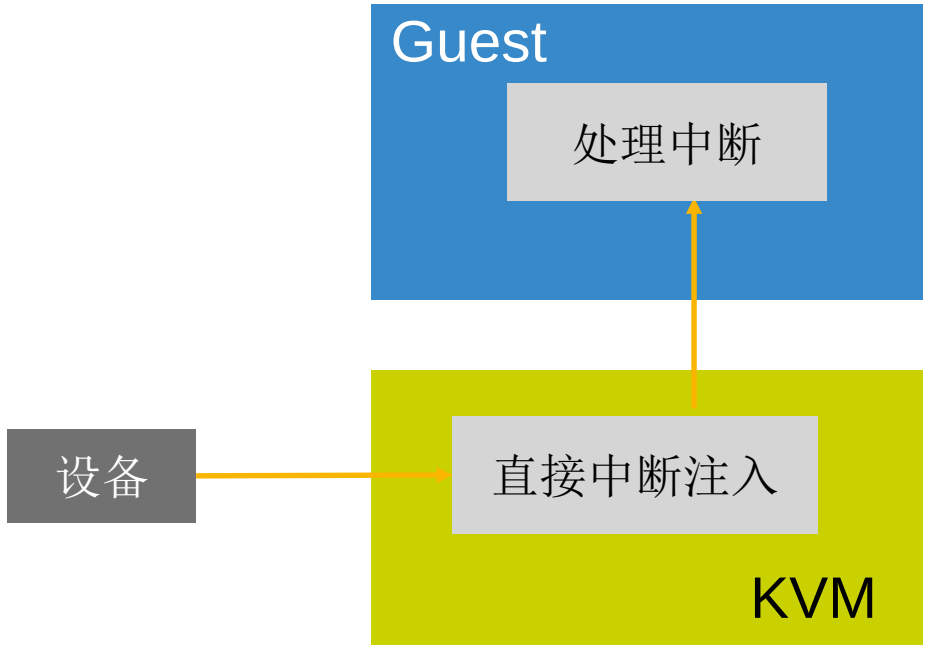


中断虚拟化


GIC (v3)
设备
中断转换服务
再分配器
CPU接口
CPU
ITS转换服务仿真
设备
分配器
再分配器
CPU接口
CPU
系统中存在一个分配器
– 需要仿真
Guest访问虚拟CPU接口的直接通道
Guest可直接：
• 屏蔽中断
• ACK中断
• EOI中断

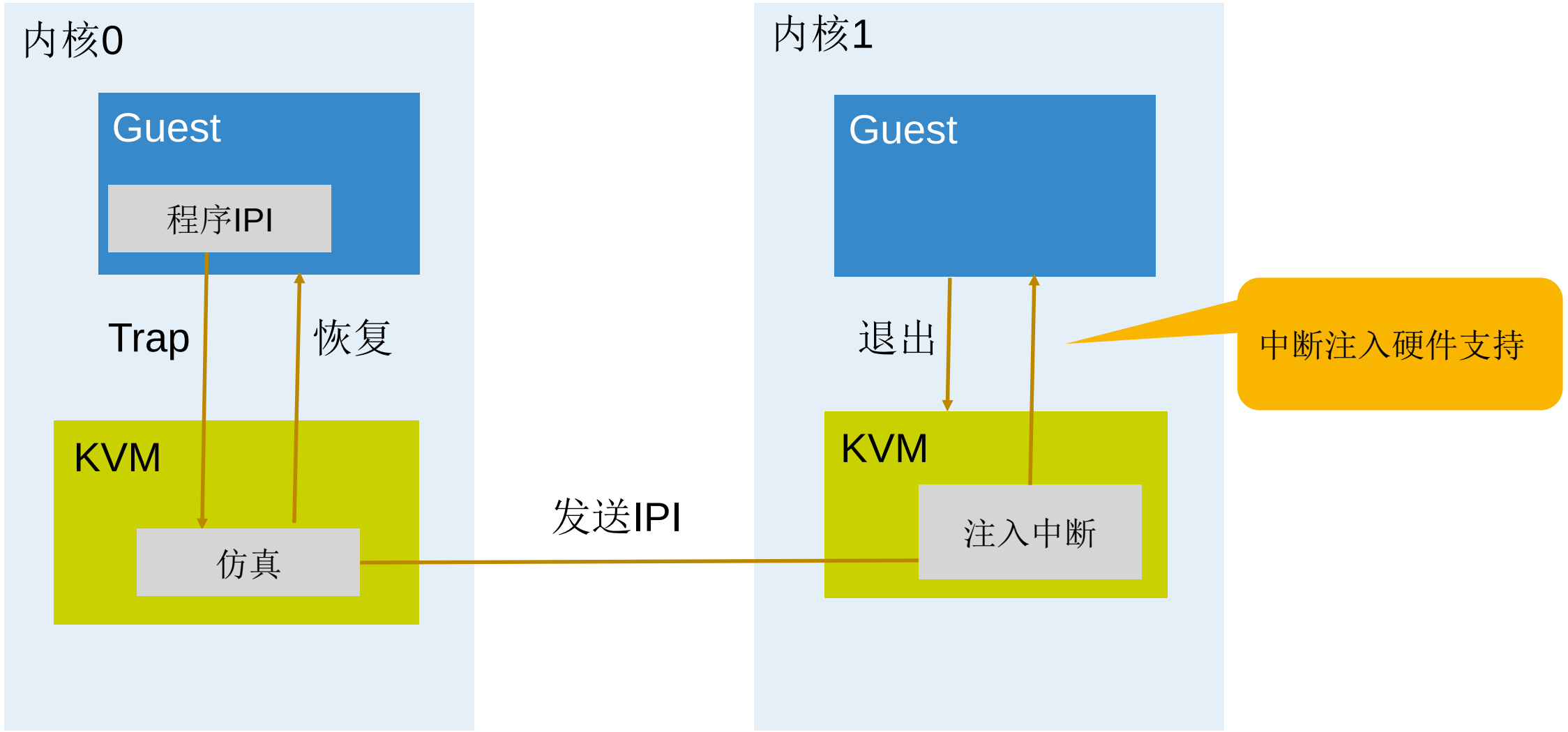
中断流





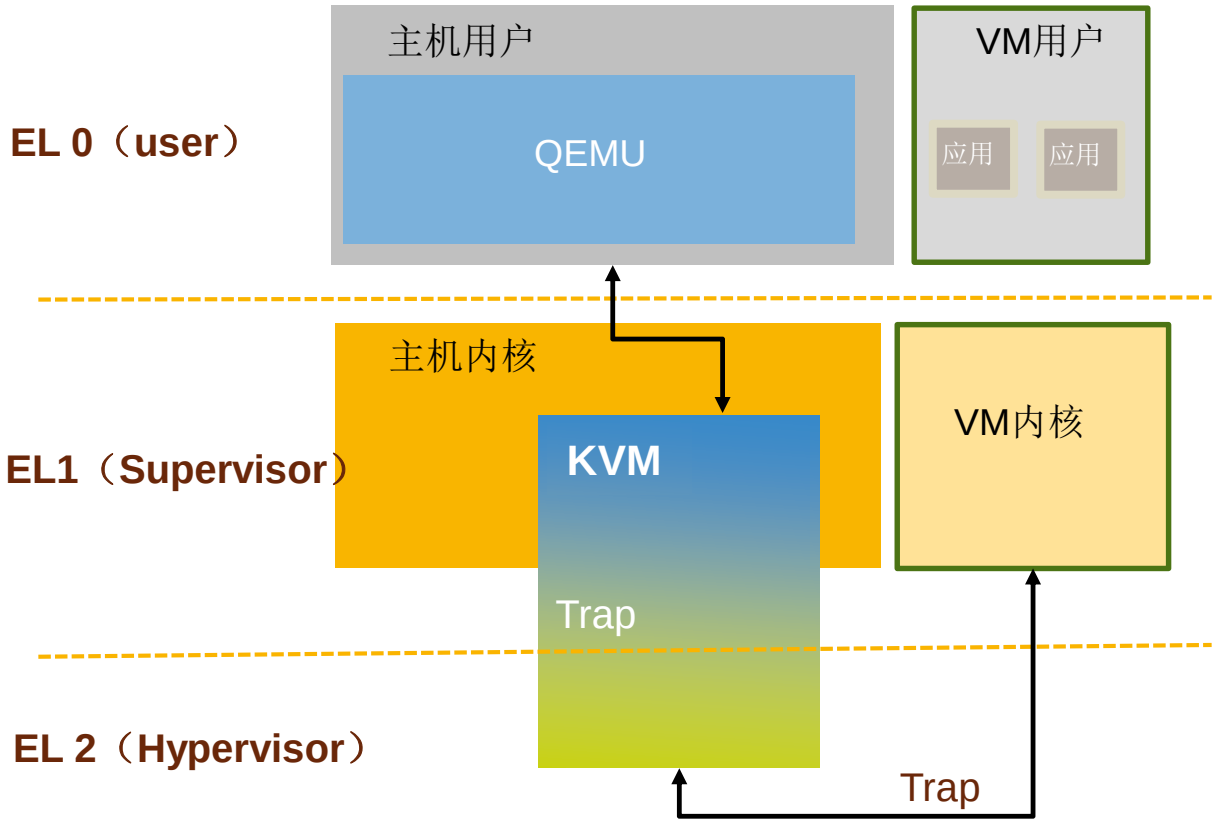
- 接收到中断后Guest被中断
- 硬件支持中断注入
- 某些情况下可能存在额外退出（但应该很少）。

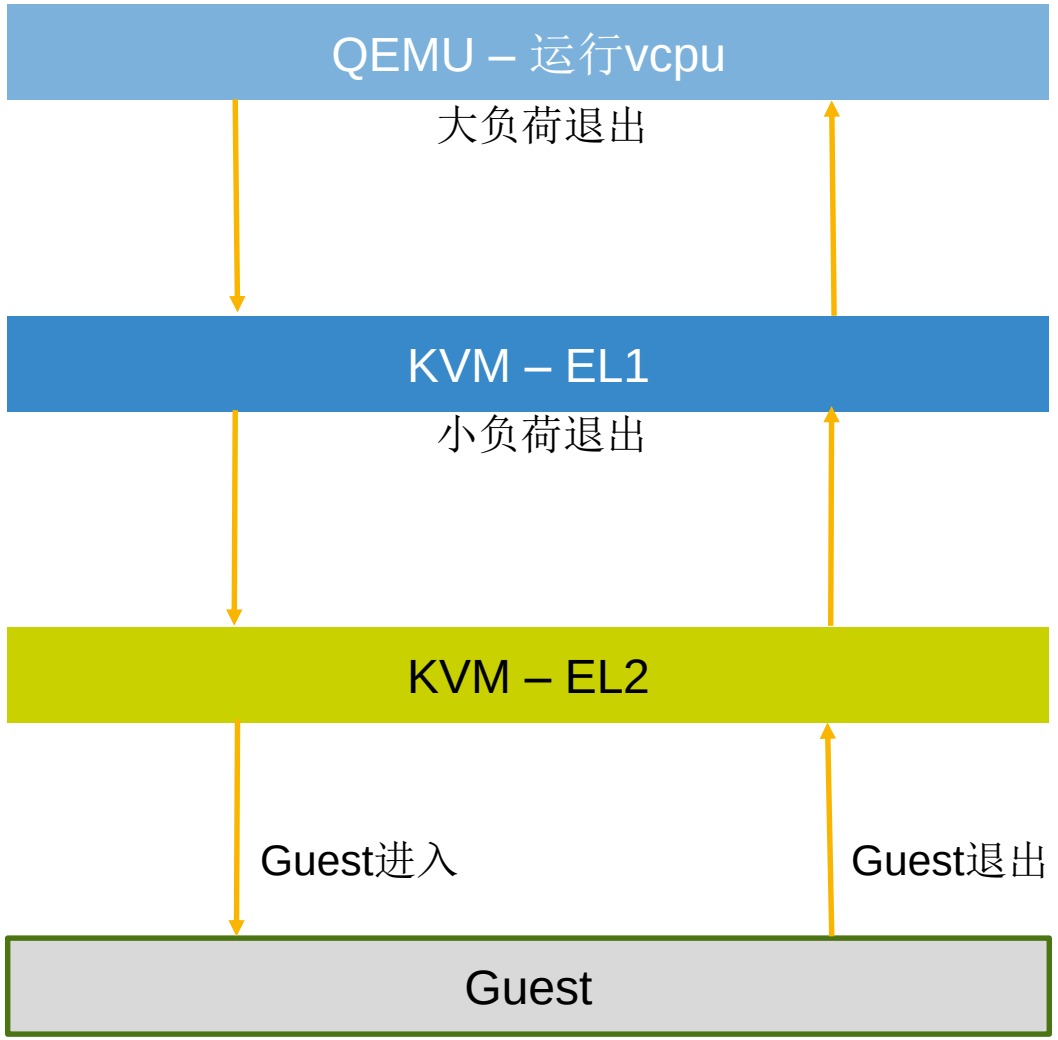
IPI流


内核0
Guest
程序IPI
Trap
恢复
KVM
仿真
发送IPI
内核1
Guest
退出
KVM
注入中断
中断注入硬件支持

ARM上的KVM

ARM上的KVM/QEMU





结果

开销来源

- 虚拟化可能附带成本：开销

- 但是在我们具有硬件扩展时是什么导致开销呢？

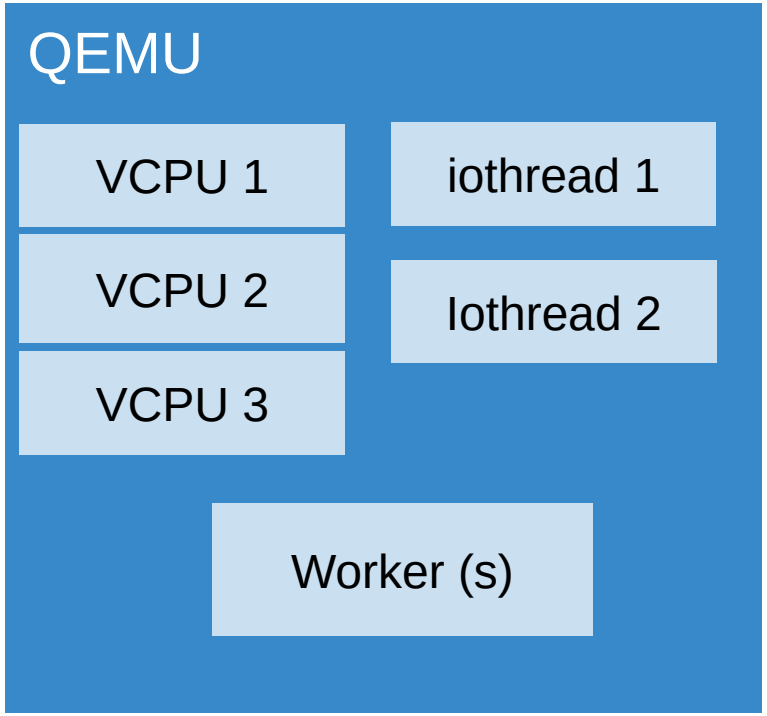
- 因Guest退出导致开销
  - 陷阱 (Trap)、中断
- Guest速度
  - 内存转换步骤更多
  - TLB/缓存污染/竞争
  - 锁竞争
- 应用延迟
  - 延迟敏感型应用在虚拟化环境中的工作特性可能不同

Guest退出 – 示例

- 退出定时框架
  - 为各种退出类型报告了在Hypervisor中花费的时间

类型	计数	最小值（cycle）	最大值（cycle）	平均值（cycle）	合计（cycle）	标准差（cycle）	计数	合计
WFX	0	0	0	0	0	0	0	0
CPC15_32	0	0	0	0	0	0	0	0
CPC15_64	0	0	0	0	0	0	0	0
CP14_MR	0	0	0	0	0	0	0	0
CP14_LS	0	0	0	0	0	0	0	0
CP14_64	0	0	0	0	0	0	0	0
HVC32	0	0	0	0	0	0	0	0
SMC32	0	0	0	0	0	0	0	0
HVC64	0	0	0	0	0	0	0	0
SMC64	0	0	0	0	0	0	0	0
SYS64	0	0	0	0	0	0	0	0
IABT_LOW	0	0	0	0	0	0	0	0
DABT_LOW	0	0	0	0	0	0	0	0
DABT_IO_MEM	0	0	0	0	0	0	0	0
DABT_USER_MEM	0	0	0	0	0	0	0	0
DABT_IO_MEM_IPI	157225	10090	57218	13385	2.104.548.054	172	15722,5	210.454.805,40
INTERRUPT	159395	4963	39654	6792	1.082.746.418	226	15939,5	108.274.641,80
TIMEINGUEST	316620	163	356454	47376	15.000.203.563	1230	31662	1.500.020.356,30
DESCHEDULED	2	7036	7036	7036	14072	0	0,2	1.407,20

KVM Benchmark考虑因素



?

- VM scaling
- 集群 (Cluster)
- QEMU线程亲和性
- CPU scaling
- 空闲/繁忙主机
- 复现性
- 中断亲和性

测试方法和分析工具

- Benchmark
  - Coremark
  - Lmbench
- 分析工具
  - 退出定时测量
  - Perf计数器（硬件计数器）

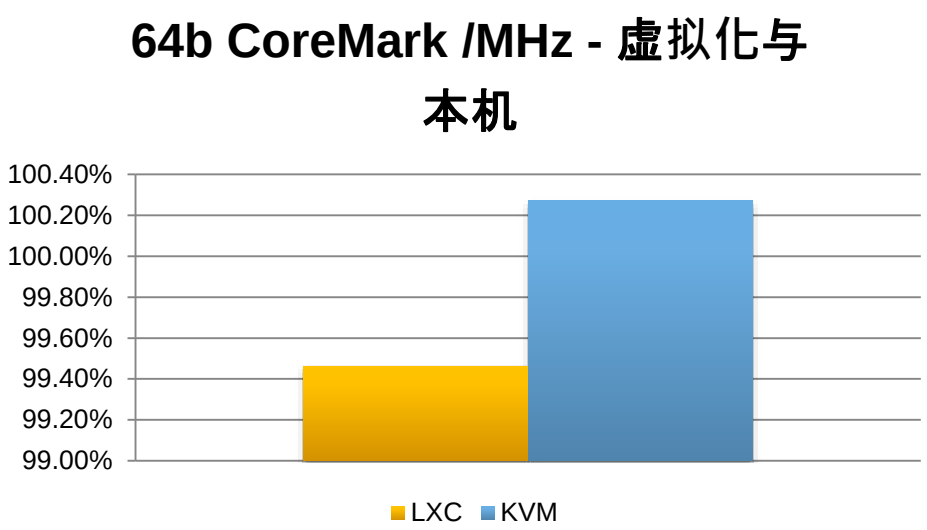
- 平台
  - LS2080 QorIQ硬件

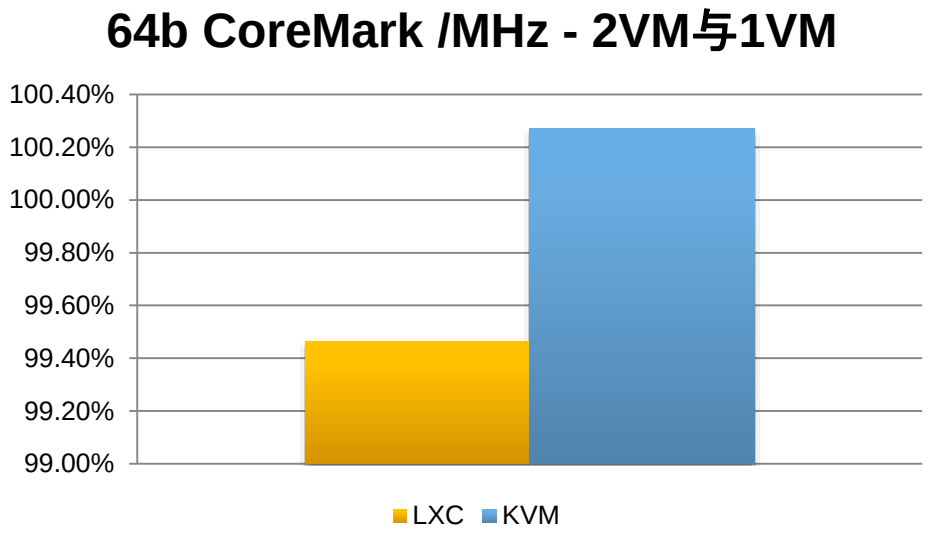
CoreMark

- Microbenchmark
- Core centric

CoreMark – 结果

guest/native[%]



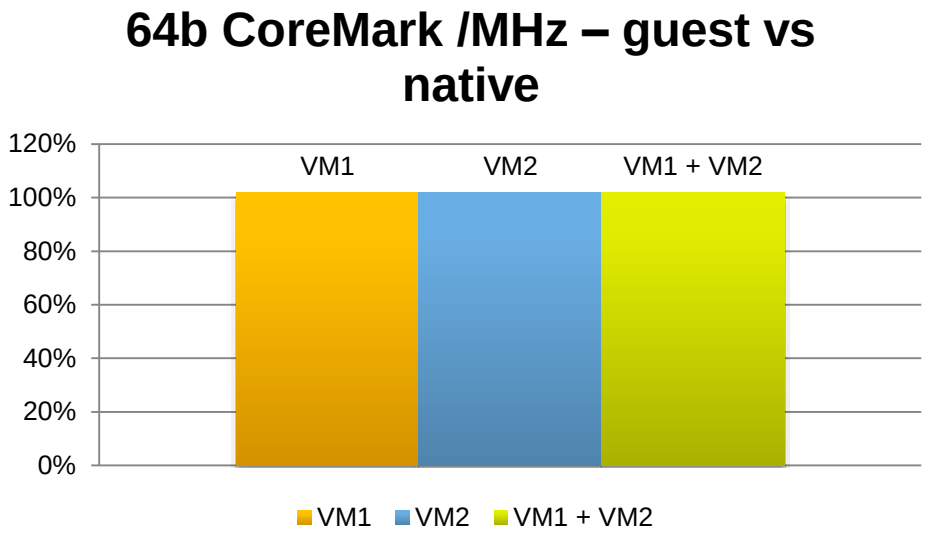


CoreMark – VM的并行性

过量预订（Oversubscription）

- Host：2个CoreMark进程在同一CPU上运行
- Guest：2个VM (VCPU)在同一CPU上运行，每个均在同一CPU上运行一个CoreMark实例

guest/native[%]



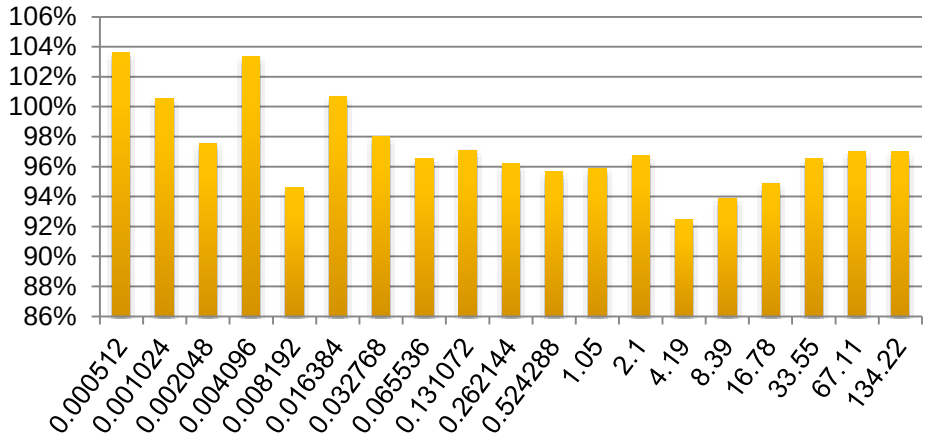
LMbench

- Synthetic microbenchmark
  - 带宽Benchmark
    - 存储器带宽
    - IPC带宽
    - 缓存I/O带宽
- Latency Benchmark
  - 存储器读取
  - 信号处理
  - 进程创建
  - 上下文切换
  - 进程间通信
  - 文件系统

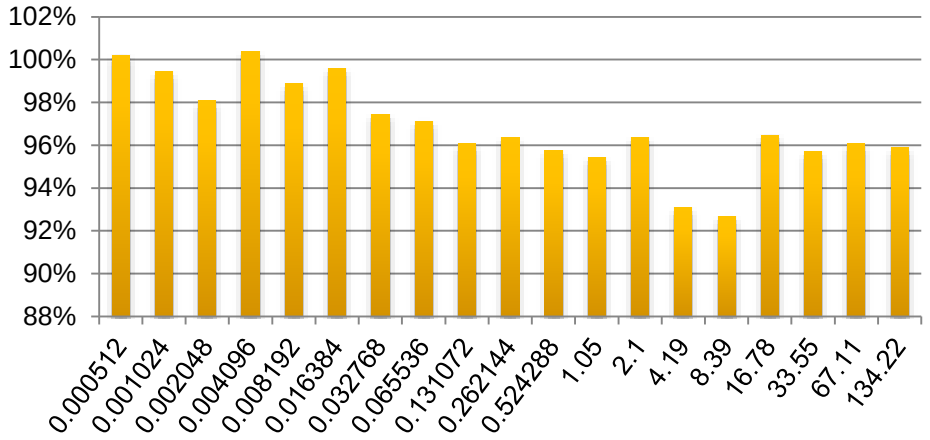
LMBench – 通信带宽

guest/native[%]

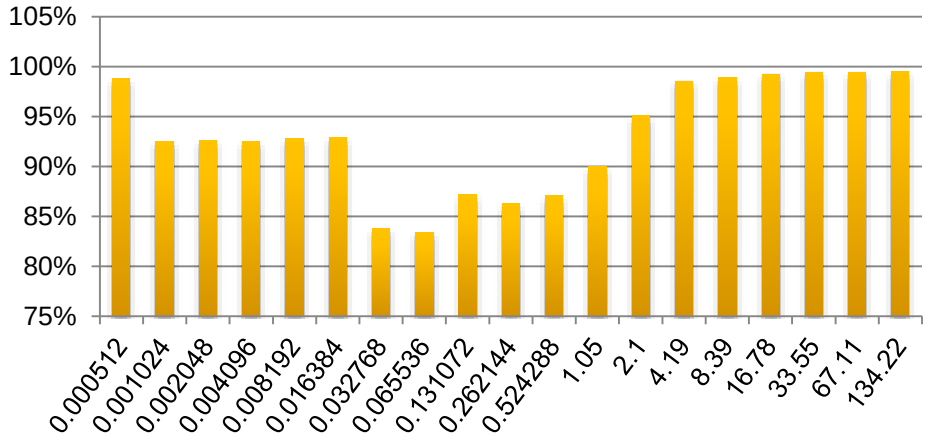
文件读取带宽



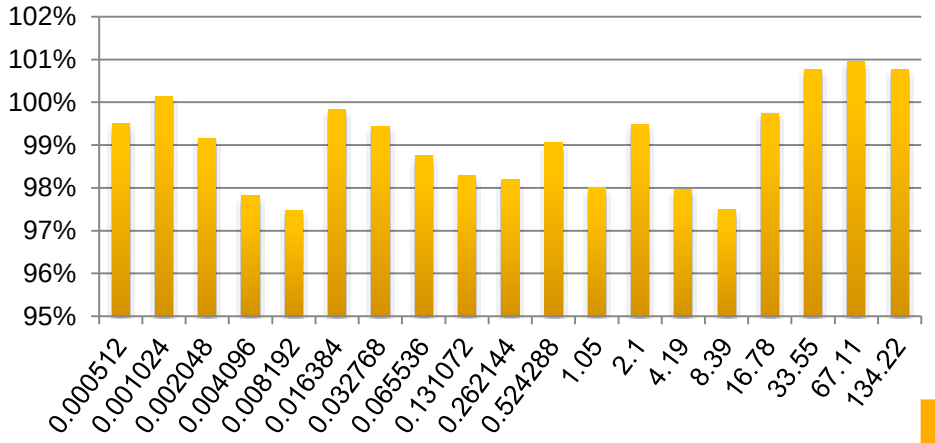
读取open2close带宽



Mmap读取带宽



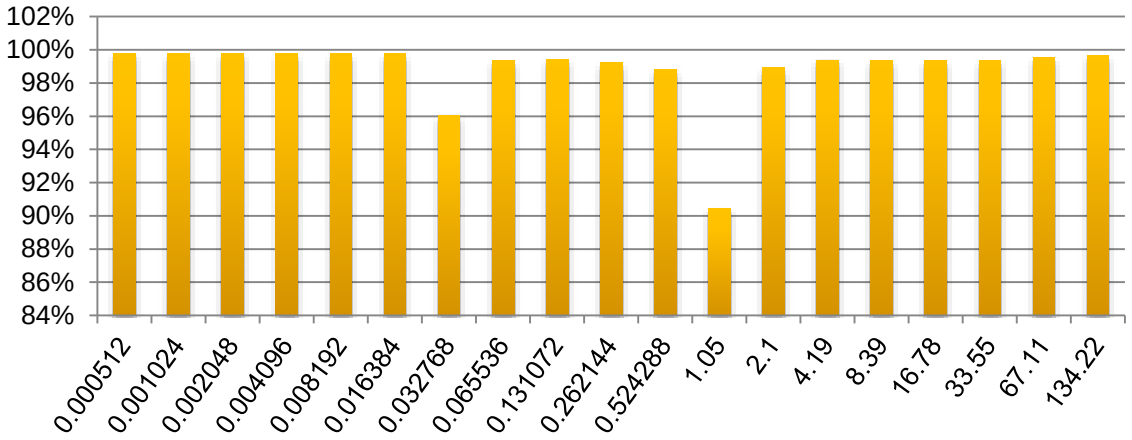
Mmap读取open2close带宽



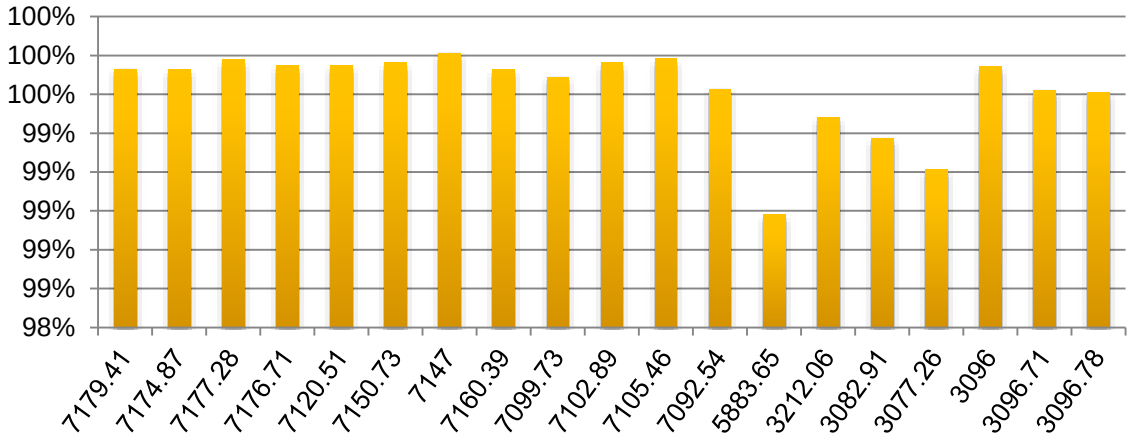
LMBench – 内存带宽

guest/native[%]

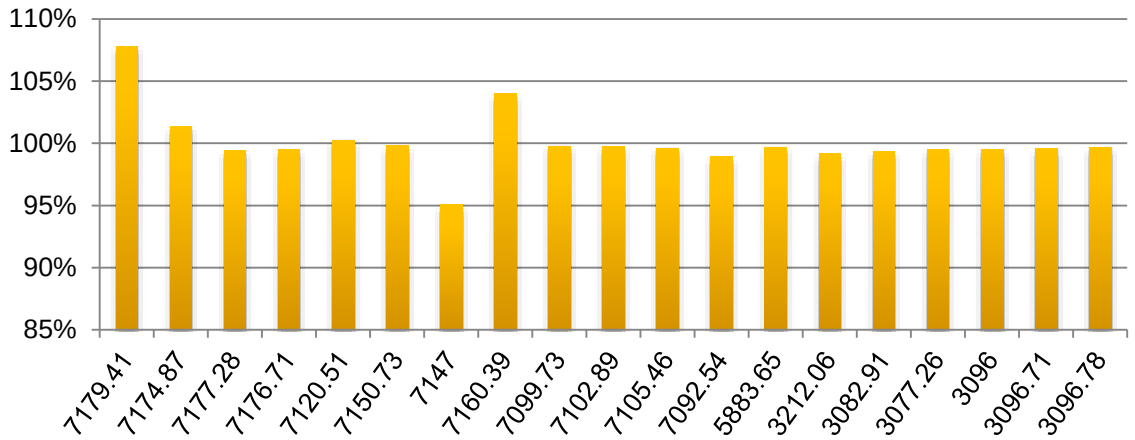
内存读取带宽



内存写入带宽

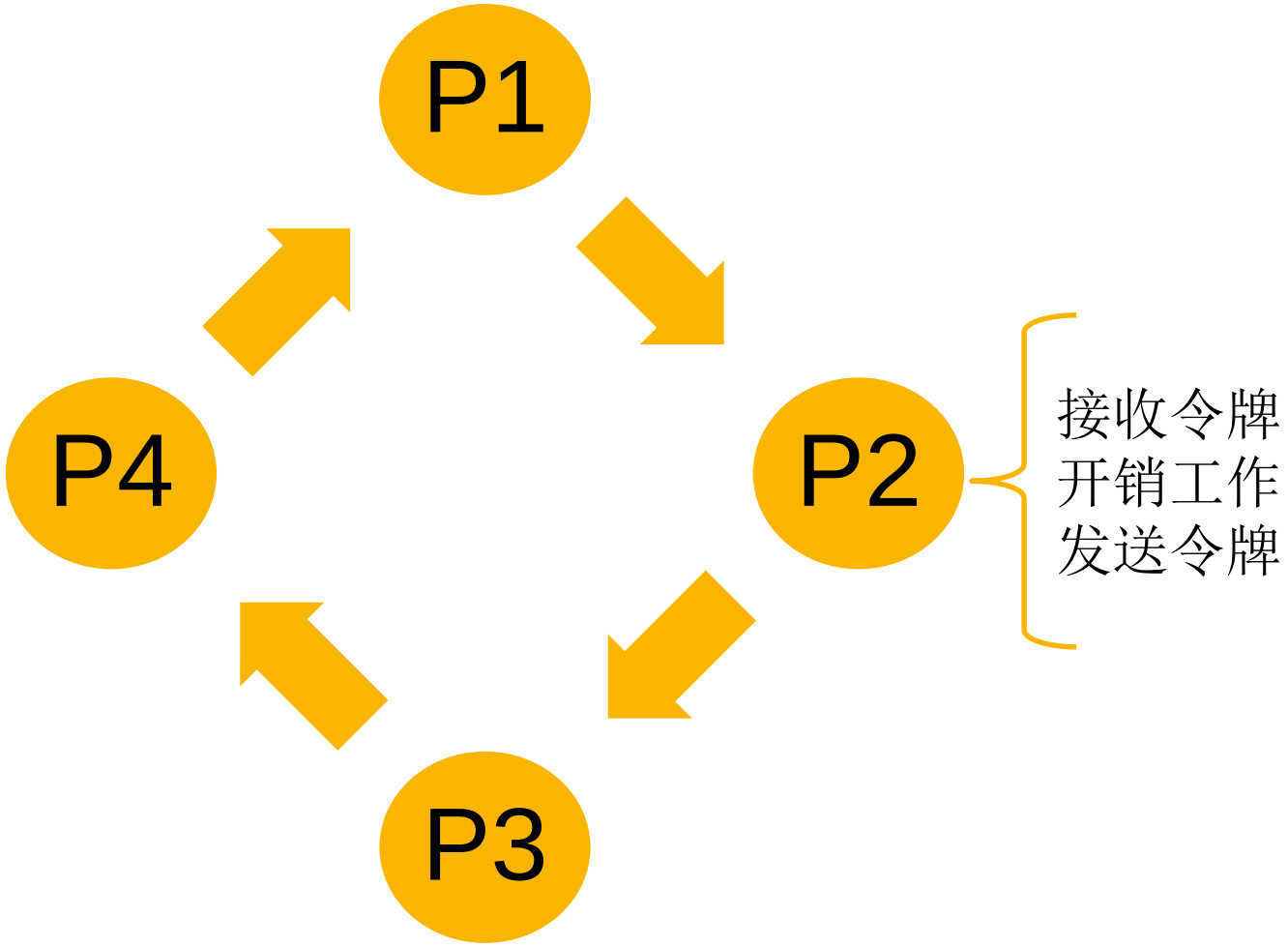


内存部分读取/写入带宽



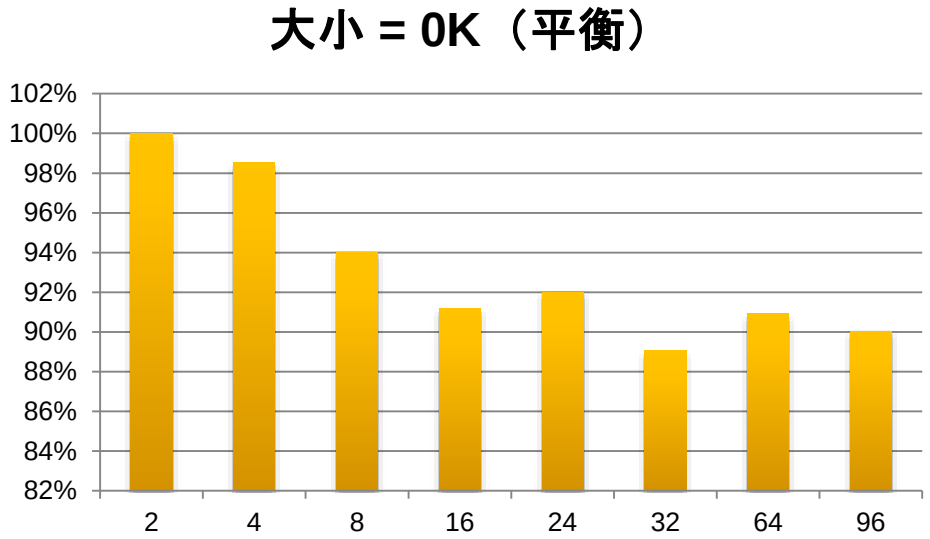
LMBench – 上下文切换 sub-benchmark

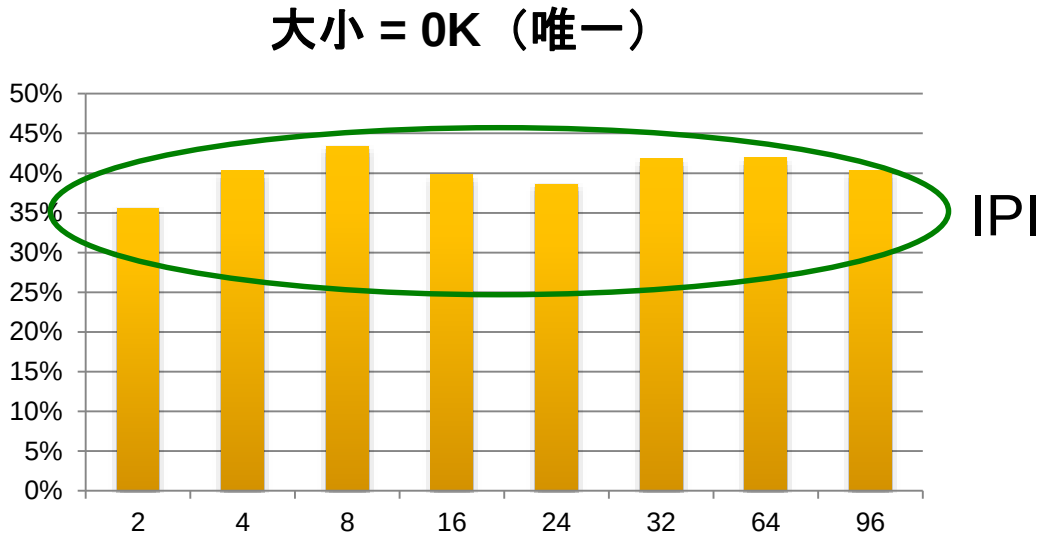
- 平衡
  - 所有进程在同一内核上
- 唯一
  - 每个进程都在不同内核上

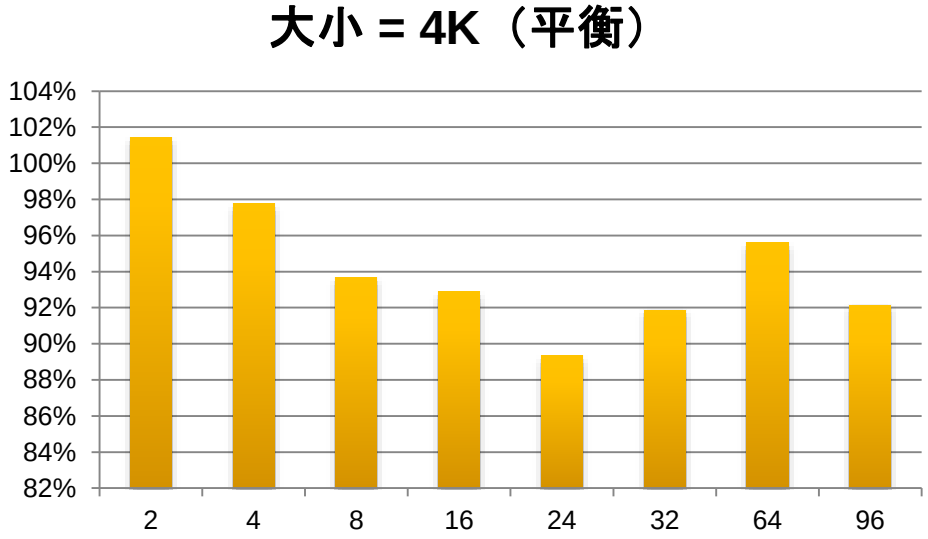


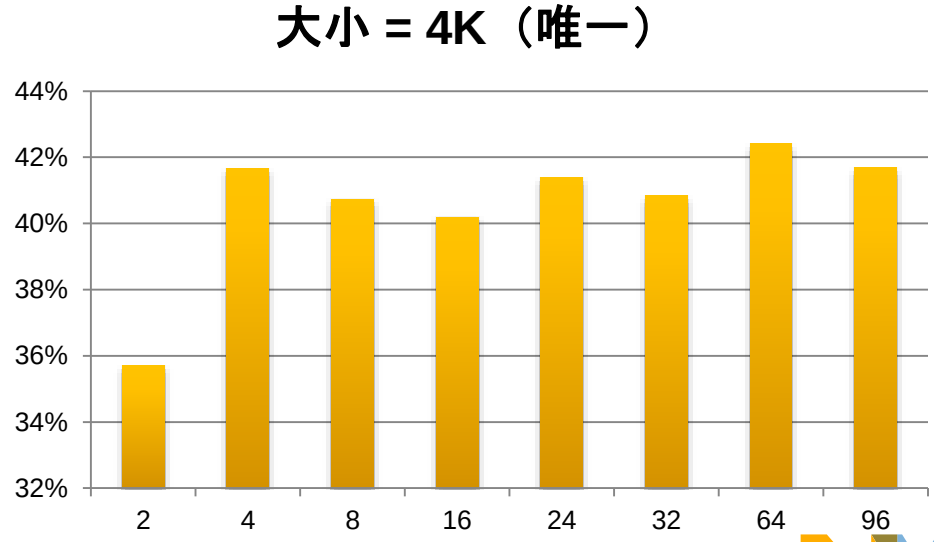
LMBench – 上下文切换延迟

guest/native[%]



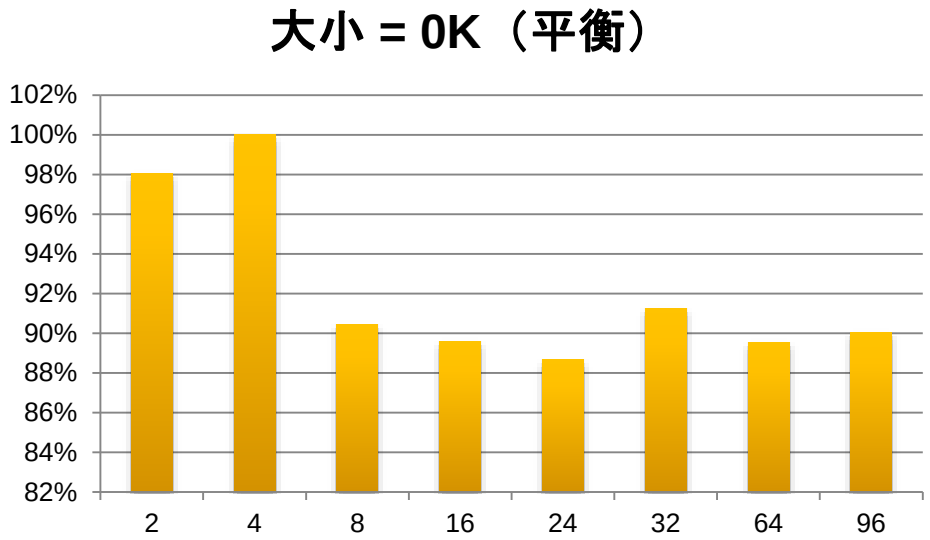


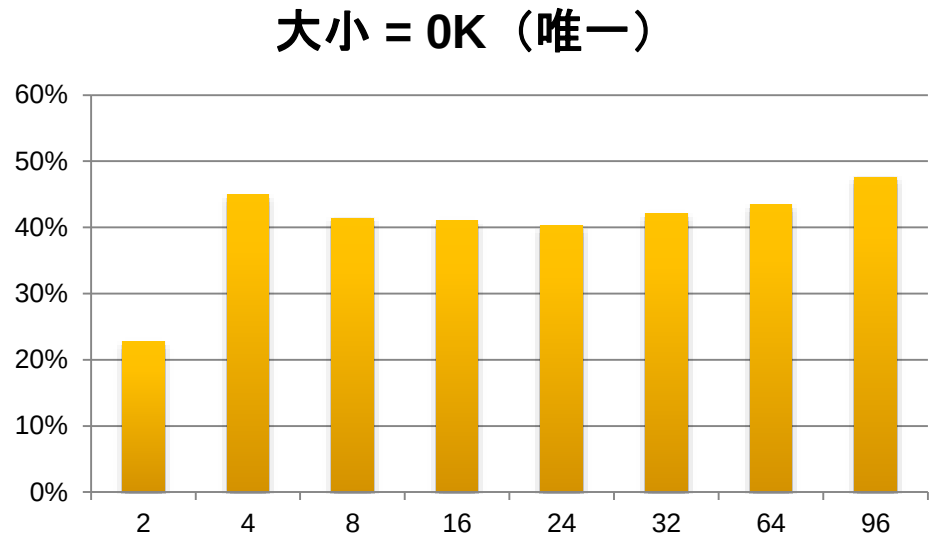


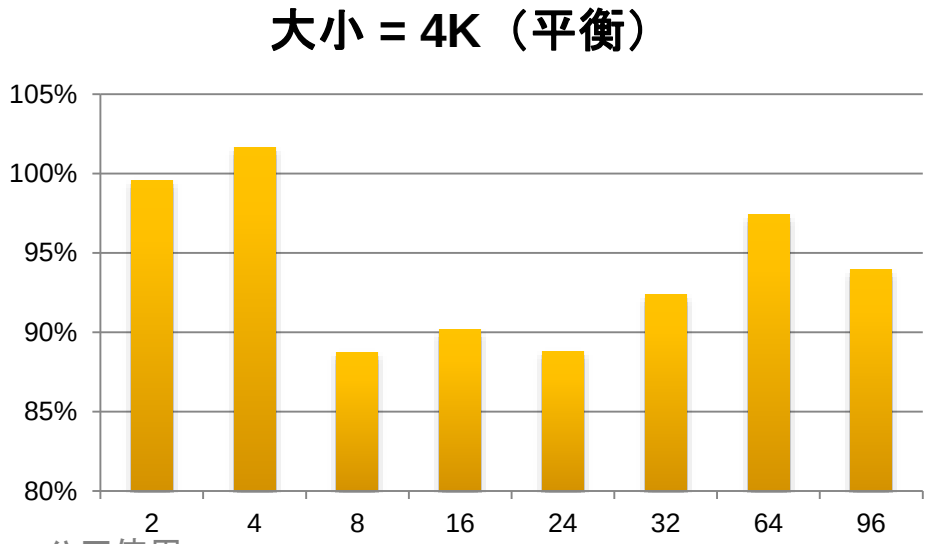


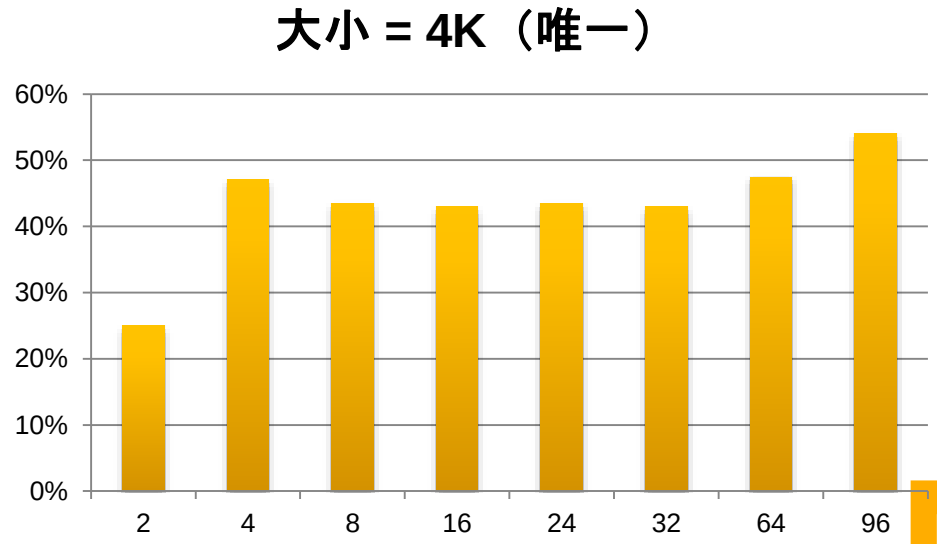
LMBench – 上下文切换延迟 – Scaling

guest/native[%]

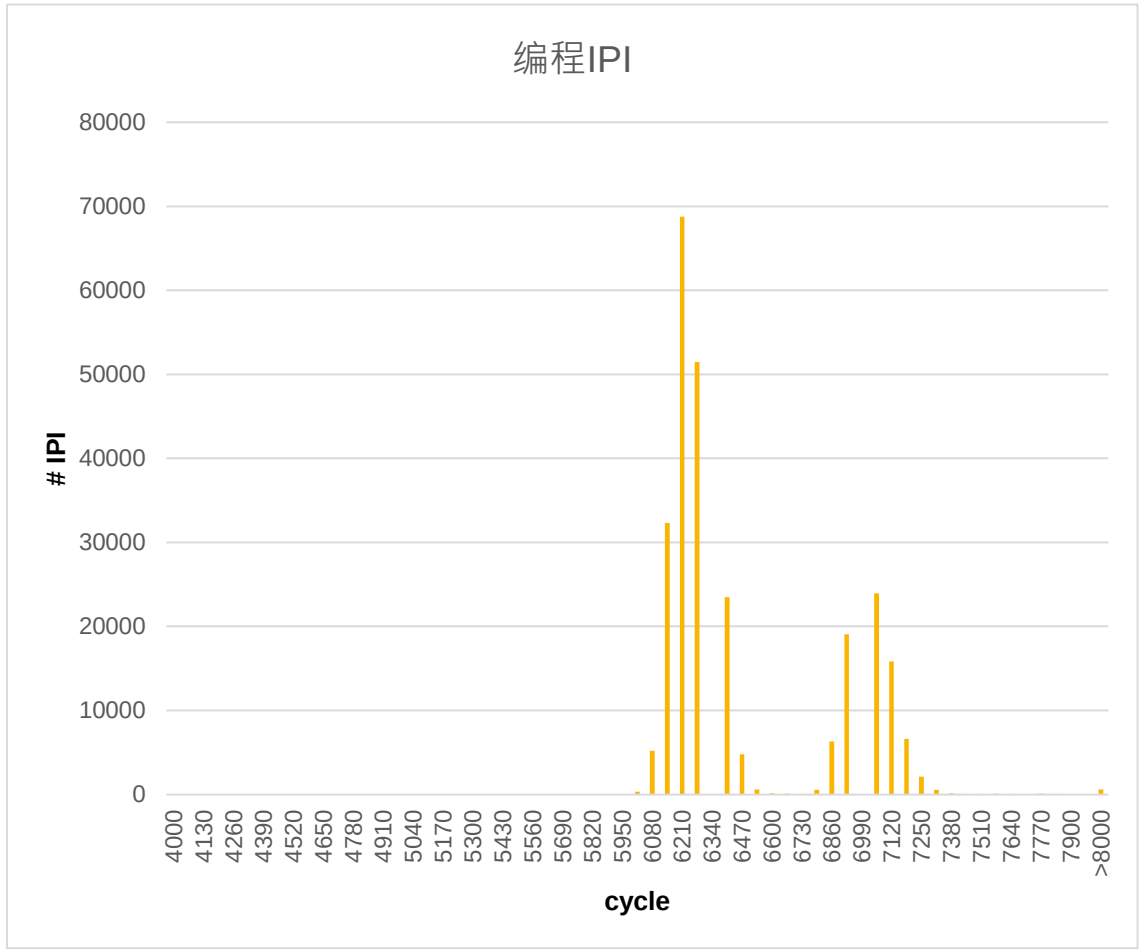


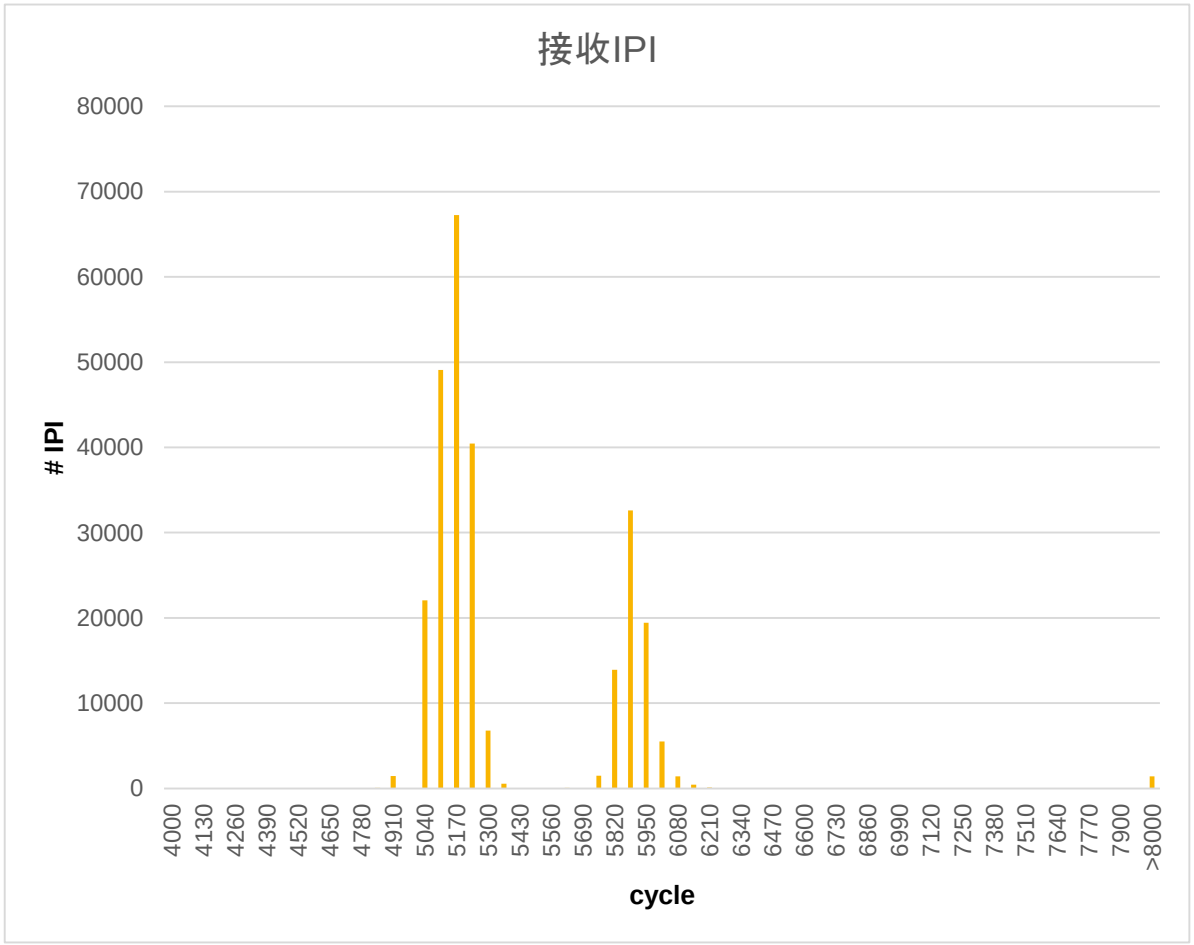




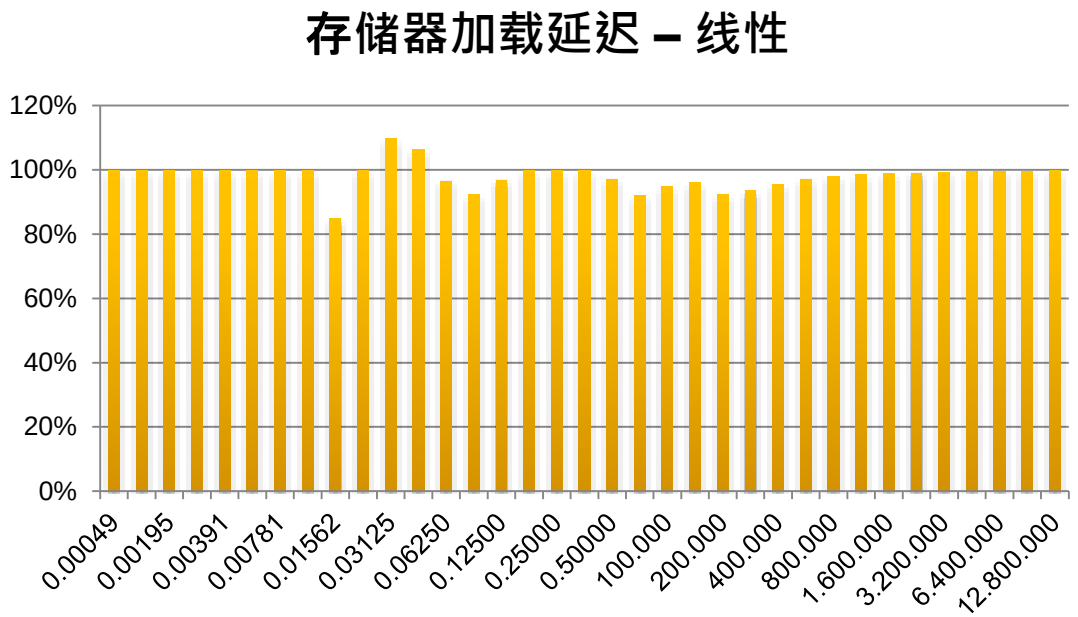


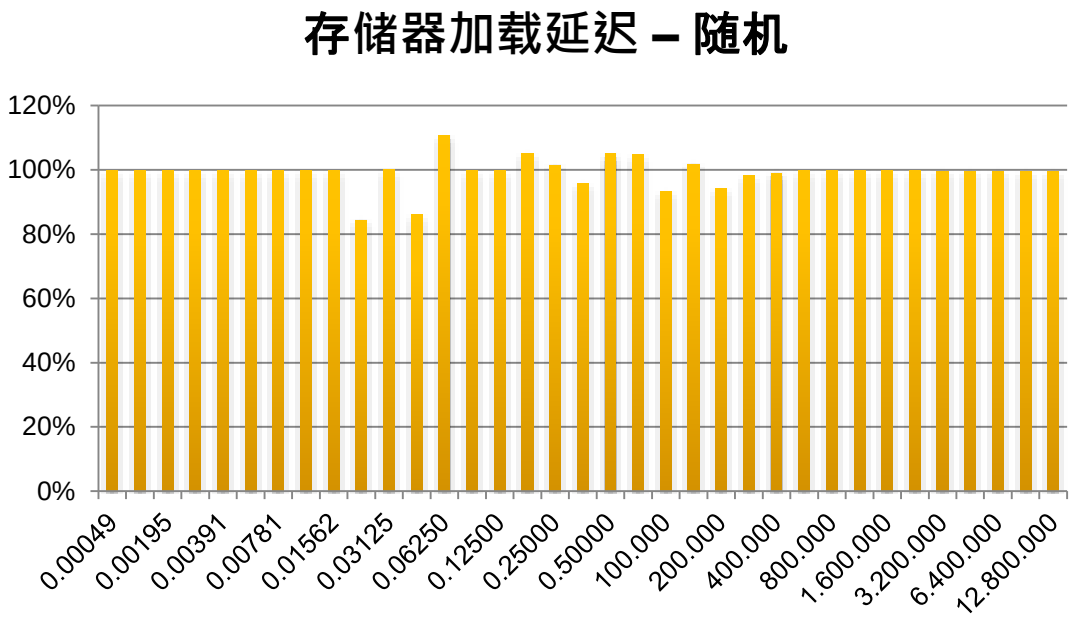
上下文切换延迟分配


编程IPI
80000
70000
60000
50000
40000
30000
20000
10000
0
IPI
4000
4130
4260
4390
4520
4650
4780
4910
5040
5170
5300
5430
5560
5690
5820
5950
6080
6210
6340
6470
6600
6730
6860
6990
7120
7250
7380
7510
7640
7770
7900
>8000
cycle


接收IPI
80000
70000
60000
50000
40000
30000
20000
10000
0
IPI
4000
4130
4260
4390
4520
4650
4780
4910
5040
5170
5300
5430
5560
5690
5820
5950
6080
6210
6340
6470
6600
6730
6860
6990
7120
7250
7380
7510
7640
7770
7900
>8000
cycle

存储器加载延迟 – VM与本机





结论

结论

- 对于与内核相关的 benchmark, 由于退出很少, 因此性能良好
- 开销来源为：特别是由IPI和中断仿真造成的Guest退出
  - 通过重新设计GIC分配器仿真而提升性能
- 与内存相关的benchmark在虚拟化环境中没有显示出重要的开销
  - 页表级数不会产生重大影响

SECURE CONNECTIONS
FOR A SMARTER WORLD

版权声明