ArchSummit

全球架构师峰会（北京）2014

阿里云虚拟化技术自研之路

阿里云 张献涛（旭卿）

自我介绍

阿里云 aliyun.com

- 张献涛，花名旭卿，毕业于武汉大学，获信息安全博士学位。
- 供职于阿里巴巴集团，负责阿里云虚拟化技术团队，主导阿里云下一代虚拟化架构的设计与研发工作。
- 加入阿里巴巴之前，供职于英特尔亚太研发中心虚拟化部门，有9年的虚拟化项目经验，先后担任高级工程师、主任工程师、虚拟化架构师等职位。
- 多个开源虚拟化项目Xen、Linux/KVM的主要贡献者，曾担任Xen项目子系统的Maintainer，并为KVM虚拟化项目增加了跨平台支持，实现了KVM在IA64平台的支持，并担任Linux内核KVM/IA64项目的Maintainer。
- 2011年，研发的HAXM虚拟机加速器为Android系统模拟器插上了飞翔的翅膀，性能提升数倍，开发效率倍增，惠及数以百万的Android应用开发人员，并因此获得英特尔最高成就奖(IAA)。
- 在国内外发表虚拟化相关论文多篇以及拥有多项美国专利。

议程

阿里云 aliyun.com

- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

议程

阿里云 aliyun.com

- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

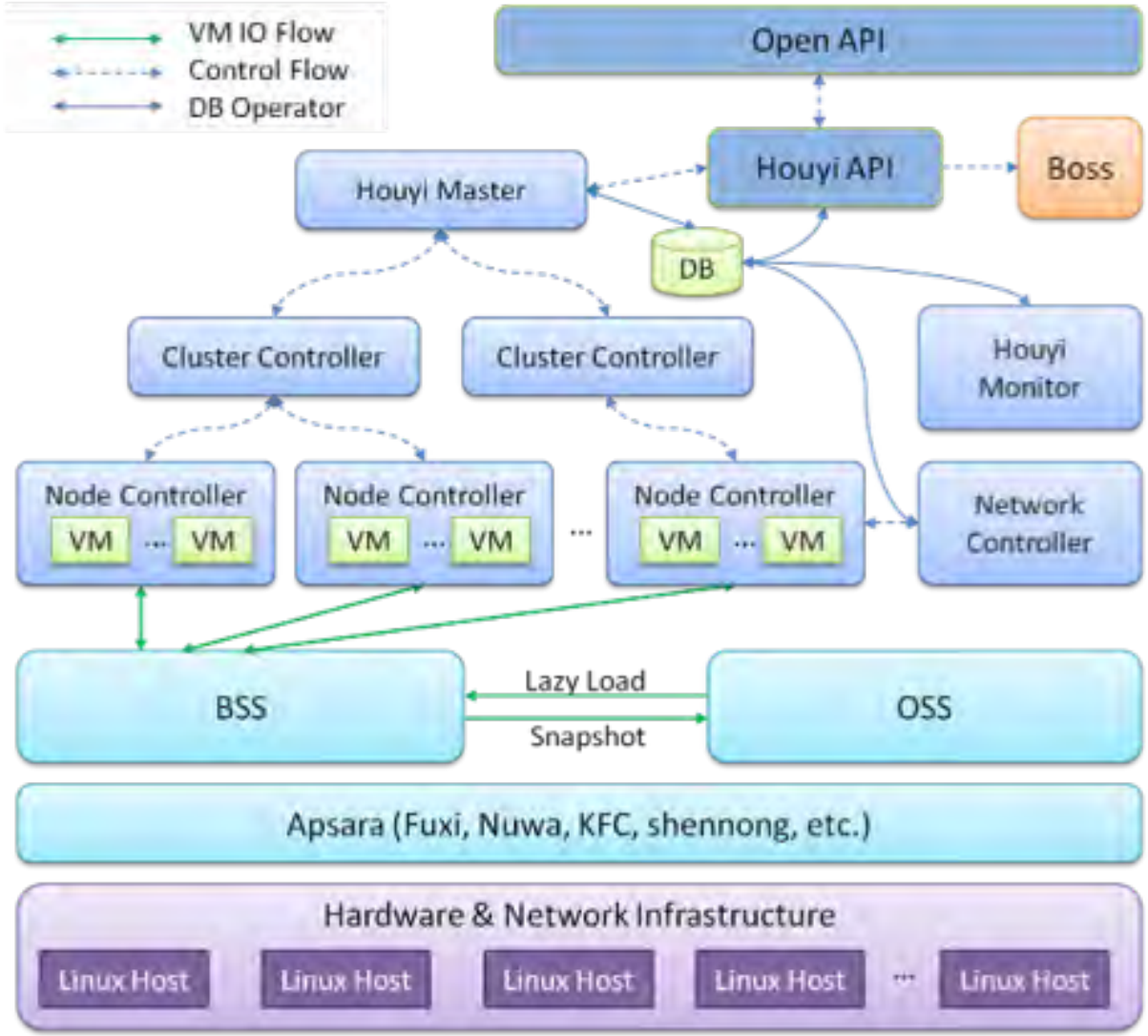
ECS产品简介

阿里云 aliyun.com

- **应用程序的基础运行环境**
  - ECS(云服务器)是阿里云产品体系中，最基础的计算服务，通常用作应用程序的运行环境，其最重要的特点是弹性。
  - 每个ECS实例上都运行着用户选择的操作系统，一般是某个Linux或Windows的发行版。用户的应用程序运行在实例的操作系统之上。
- **弹性的伸缩能力**
  - ECS的最重要的特点是弹性，支持垂直和水平扩展两种能力。垂直扩展，可以在几分钟内升级CPU和内存，实时升级带宽；水平扩展，可以在几分钟内，创建数百个新的实例，完成任务后，可以立刻销毁这些实例

ECS系统架构图

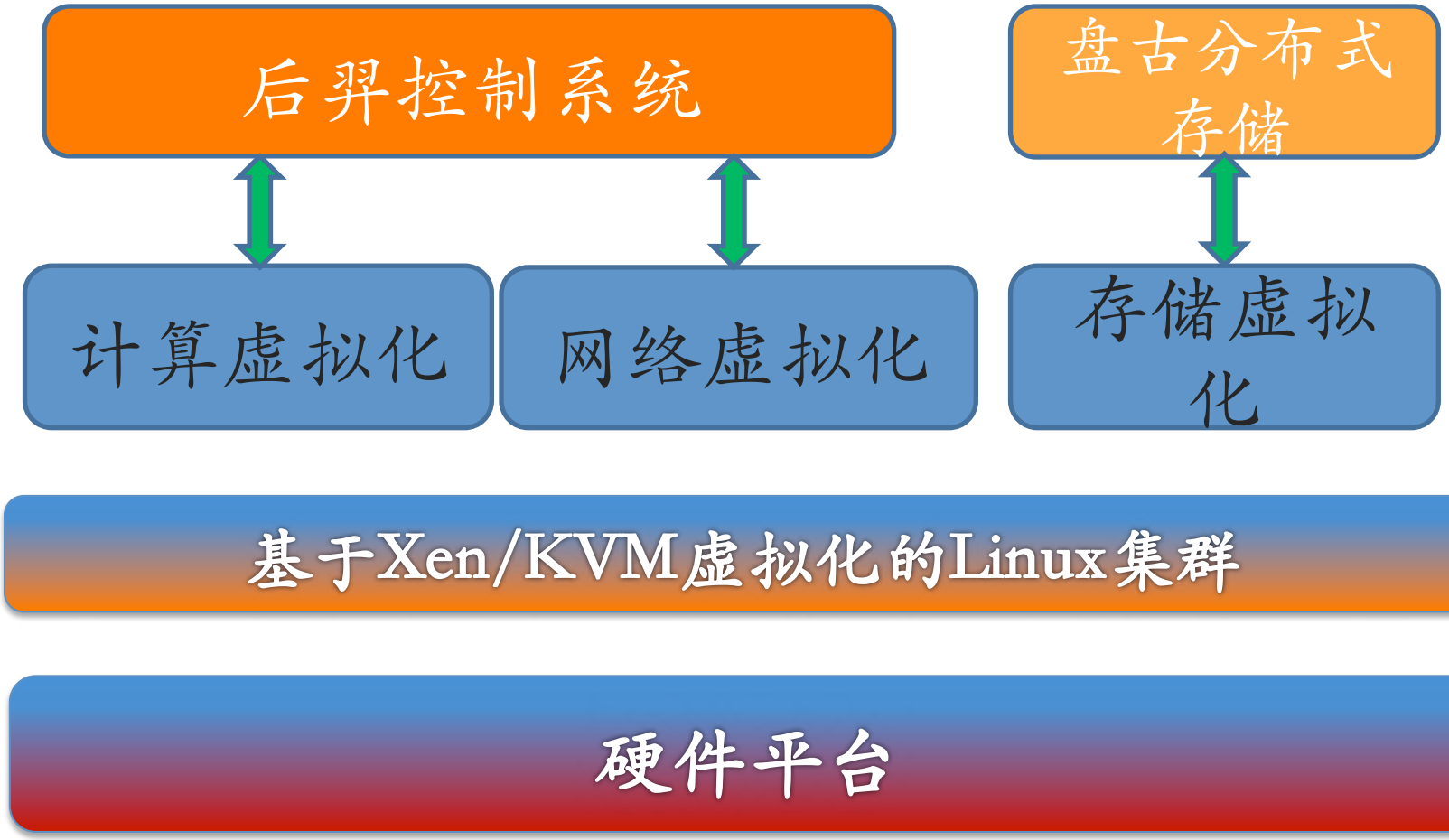
- 虚拟化支持
- 分布式文件存储
- 快照制作
- 快照回滚
- 自定义image
- 故障迁移
- 在线迁移
- 网络组隔离
- 防ARP欺骗
- 自定义防火墙功能
- 支持防DDos攻击
- 提供流量清洗服务
- 动态升级



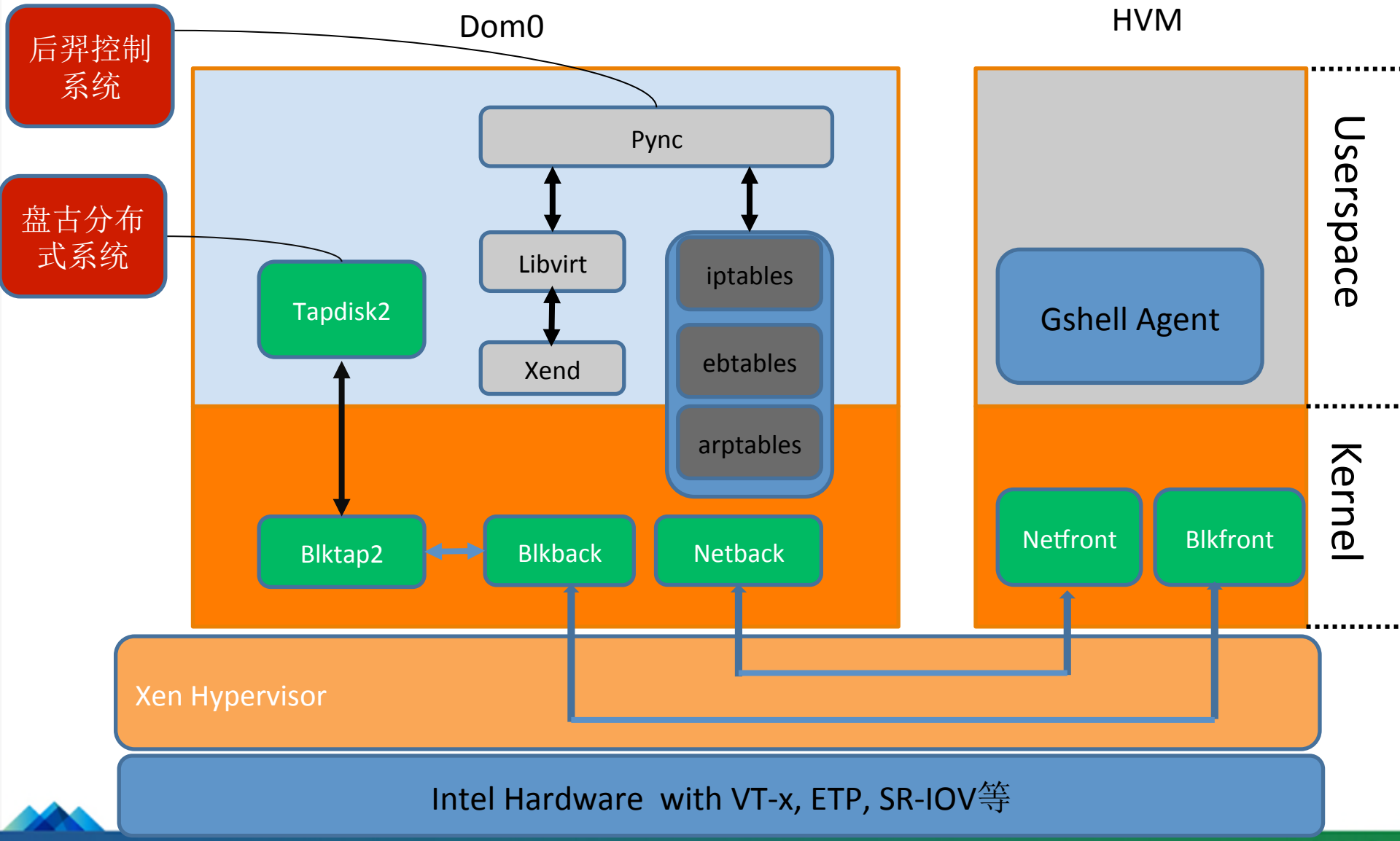
议程

- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

ECS软件系统架构


后羿控制系统
盘古分布式存储
计算虚拟化
网络虚拟化
存储虚拟化
基于Xen/KVM虚拟化的Linux集群
硬件平台

ECS虚拟化底层架构


后羿控制系统
盘古分布式系统
Dom0
HVM
Pync
Libvirt
Xend
Tapdisk2
iptables
ebtables
arptables
Gshell Agent
Userspace
Kernel
Blktap2
Blkback
Netback
Netfront
Blkfront
Xen Hypervisor
Intel Hardware with VT-x, ETP, SR-IOV等

ECS虚拟化软件模块

- **Hypervisor 虚拟层(Including Xen , Xen Tools, Xend等)**
  - 基于成熟的开源软件Xen
  - 为优化性能和稳定性，Xen核心代码改动超过100+项
  - 为增加系统多样性，基于KVM的其它Hypervisor方案在研
- **Dom0 内核**
  - 基于Ali 内核分支，独立研发
  - 涉及700+多个内核改动
- **高性能前后端通讯技术（ PV Driver)**
  - 基于开源的PV Driver进行研发优化
  - 优化后的高性能Driver提供更稳定高性能服务，优化项达近20项
  - 虚拟化网络驱动最高支持300W+ PPS

ECS虚拟化关键技术

阿里云 aliyun.com

- 硬件虚拟化技术
  - CPU采用硬件虚拟化技术VT-x， 内存采用EPT方式
  - 基于硬件虚拟化技术，VM性能可达同规格物理机95%以上
- 热迁移技术
  - 底层基于Xen热迁移研发，改动超过20+项
  - 独立研发热迁移控制系统
  - 优化后的热迁移达到业界领先水平
- 内核Hotfix技术
  - 独立研发AliHotfix技术
  - 独立研发Hypervisor Hotfix技术，独具创新型

议程

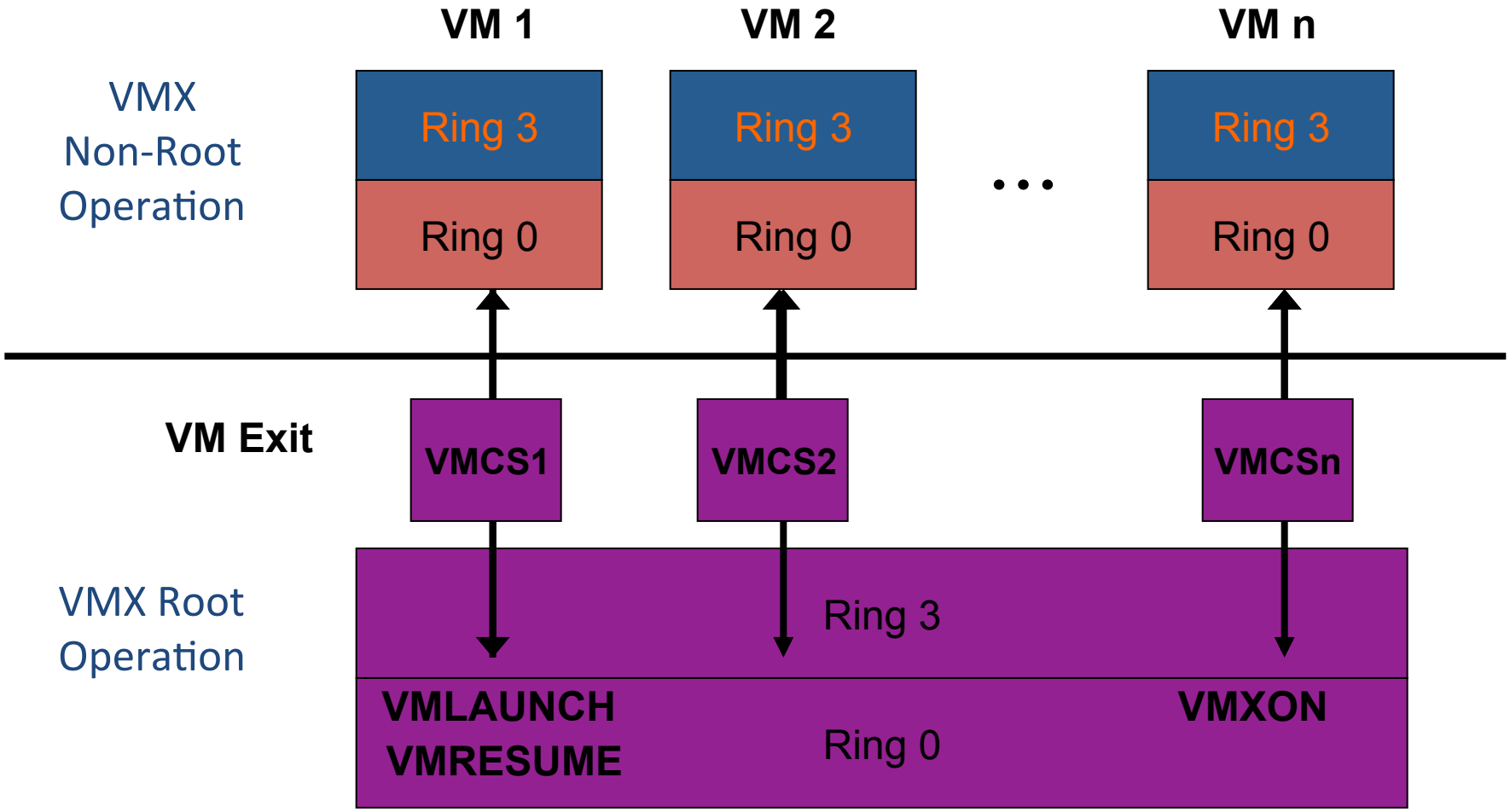
- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

Intel® Virtualization Technology

- A hardware-assisted virtualization technology, named as Intel® Virtualization Technology, or Intel® VT
  - For Intel® 64, VT-x
  - For directed I/O, VT-d
  - For connectivity, VT-c
- VT-x: A new form of Intel® 64 CPU operation
  - Provides mechanisms for VMM software control of Intel® 64 CPUs
  - Includes “hooks” necessary for software control of memory and I/O resources
- VT-d: An extension of chipset technology for directed I/O
  - DMA remapping
  - Interrupt remapping etc.
- VT-c: A collection of I/O virtualization technologies
  - Lower CPU utilization
  - Reduced system latency
  - Improved throughput

ECS全面支持Intel VT技术提升计算性能

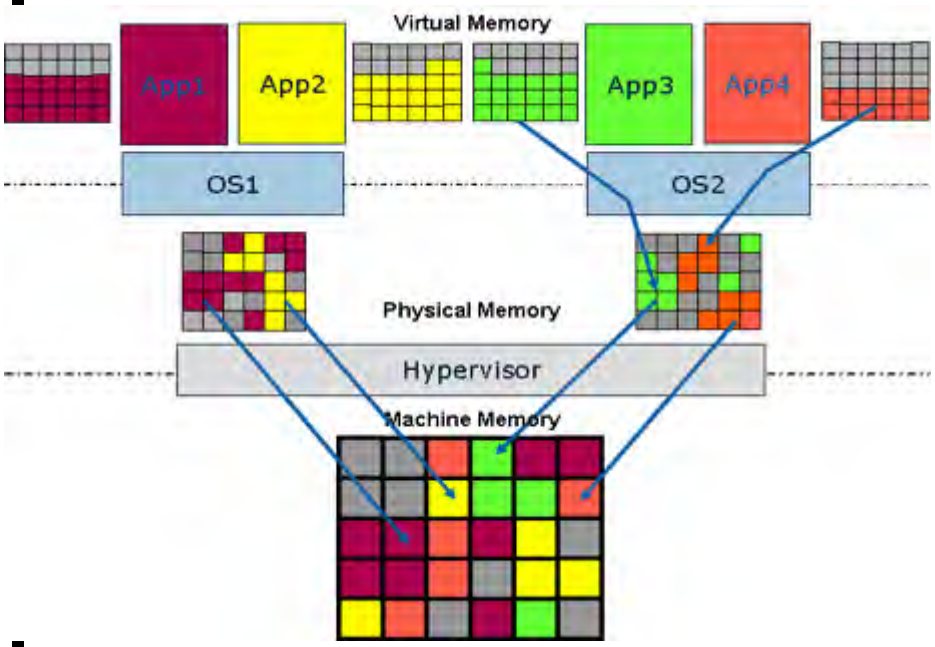
Intel VT-x CPU硬件虚拟化技术



VT-x技术让ECS CPU虚拟化效率提升至物理机95%以上

Intel Extended Page Tables(EPT)

- EPT是为内存虚拟化而生
  - 降低软件复杂度
  - 提高内存虚拟化效率
- EPT让CPU感知两层页表
  - 完全消除内存虚拟化的软件参与
  - 客户机页表由硬件直接使用
- VM自由控制自己页表
  - 减少大量的VM Exit事件
  - 减少由于影子页表等机制造成的内存浪费
  - 内存虚拟化逻辑复杂度大大降低

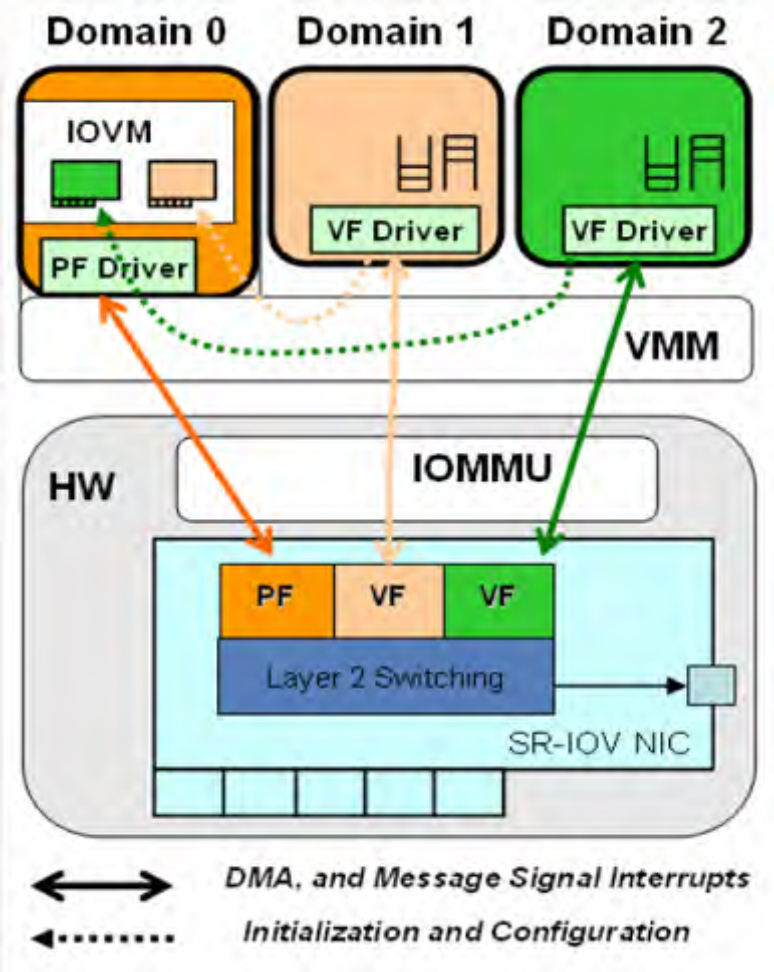


- **CPU TLB ‘walks’ Virtual memory to Physical memory**
- **EPT ‘walks’ Physical memory to Machine memory**

EPT使VM内存访问效率<70%提高至97%以上

IO硬件虚拟化技术（SRIOV）

- One PF (Physical Function) can support multiple VF
  - Physical Function is a normal PCIe function support SR-IOV capability
  - Virtual Function is light-weight PCIe function that can be accessed by guest directly
    - Resource sharing among VF like ATC, configuration etc
    - VF own non-shared resource for function-specific service, like data buffer, working queue etc
- VF is discovered and configured by VMM before passing to guest
- Configuration to VF under control of PF and VMM

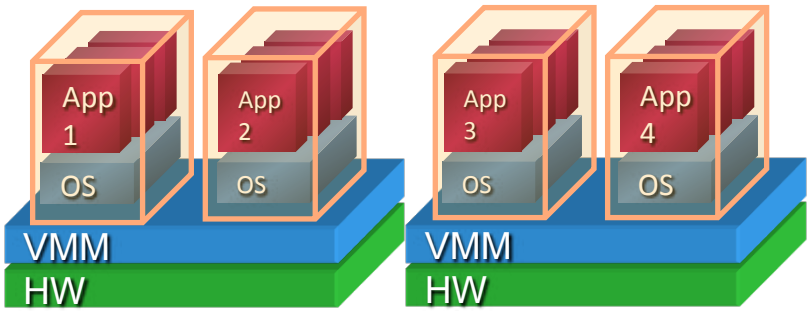


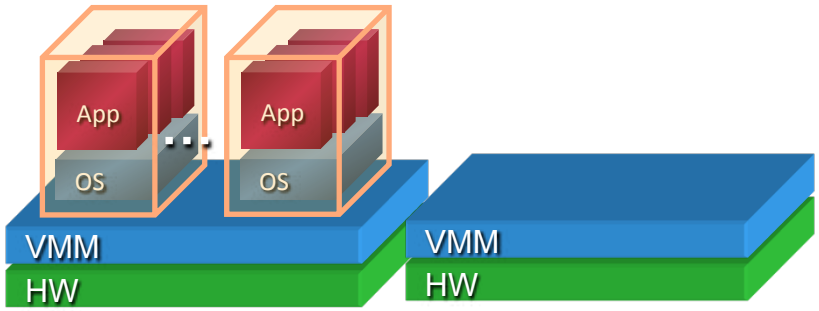
ECS使用SRIOV技术提供网络硬件级别Qos

议程

- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

ECS的典型业务场景





动态的热点均衡场景

灾难恢复

虚拟机热迁移技术

阿里云 aliyun.com

- 热迁移定义
  - 在不同物理机之间在线迁移虚拟机实例
  - 做到VM内的业务基本无感知
- 热迁移技术应用场景
  - 集群内的负载均衡
  - 机器硬件故障修复
  - 线上系统软件修复
  - 过保机器替换
  - 绿色计算
  - 主动运维

热迁移面临的技术挑战

阿里云 aliyun.com

- 线上运维标准极高
  - 要求VM Downtime控制在毫秒级
  - 网络链接无中断
  - 存储无感知
- 线上系统的复杂性
  - 镜像多样，机器型号复杂
  - 无法在线升级hypervisor, dom0
  - 历史遗留问题较多
- 虚拟化层热迁移不成熟
  - 虚拟化层Bug较多
  - Tool stack层热迁移算法和流程设计不合理
  - Qemu问题也较多

热迁移面临的技术挑战（续）

- Guest内核及PV driver支持不足
  - Debian, ubuntu等内核问题较多
- 存储层面
  - Pangu分布式存储系统
  - 锁争抢
  - Cache刷新
- 网络层面
  - 线上网络环境比较复杂
  - 各种型号交换机
  - MAC, ARP
  - SLB，VPC等

热迁移增强

阿里云 aliyun.com

- 修复虚拟化层面的一系列问题
  - Centos中断风暴问题
  - Windows双鼠标光点问题
  - ubuntu1204 2059年时间漂移问题
  - ubuntu1204 3500次迁移失败一次问题
  - VNC端口绑死问题
  - RDTSC模拟引起的性能问题
  - 解除Downtime和VM 内存大小的绑定
- 修复网络层面的多个问题
  - 解决了i350网卡问题
  - 解决了mac漂移导致的交换机封端口问题
  - 解决了某型交换机在迁移场景下的bug
  - 解决了vm迁移后fake arp网络不通问题
  - 解除网络Breaktime和VM内存大小的绑定
  - ……
- 存储层面
  - 解决了锁争抢问题: chunksweep, snapshot
  - 解决热迁移vm downtime过长的问题

热迁移优化后的指标

阿里云 aliyun.com

- 满足运维的所有条件，达到业界领先水平
  - 热迁移导致的VM宕机率 < 3%%
  - VM Downtime < 700ms
  - 网络链接无中断，网络无响应时间<2S
  - 线上机器的可迁移率 > 85%
  - 迁移过程中性能波动 <15%
  - 和Xenserver等比较，具有较强的优势
- 持续优化手段和目标
  - Multi-threading 传输
  - 引入内存压缩算法及算法的场景优化模式
  - 进一步降低迁移需要的时间
  - 进一步提高迁移成功率
  - 增加容错，降低迁移引起的宕机
  - 解决异构迁移

议程

- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

ECS Hotfix 技术

- 系统Hotfix对业务运维的意义
  - 软件系统存在Bug在所难免
  - 宕机修复引起业务中断
  - 在云环境中，物理机重启影响面更广
- 系统Hotfix的目标
  - 用户无感知修复，一切尽在不言中
  - 无需宕机，增强系统的可用性
- ECS Hotfix技术分类
  - Xen Dom0 内核 Hotfix技术
  - Xen Hypervisor Hotfix技术
  - 客户机内核的Hotfix技术

Hotfix技术是规模化业务运维立命之本

Xen Dom0 内核Hotfix技术

- 业界较成熟的内核Hotfix方案
  - Ksplice by Oracle
  - Kgraft by Novell
  - Kpatch by Redhat
- 采用自主研发的AliHotfix技术
  - 修复Dom0内核Bug
  - 修复PV 驱动Bug
  - 修复系统安全漏洞

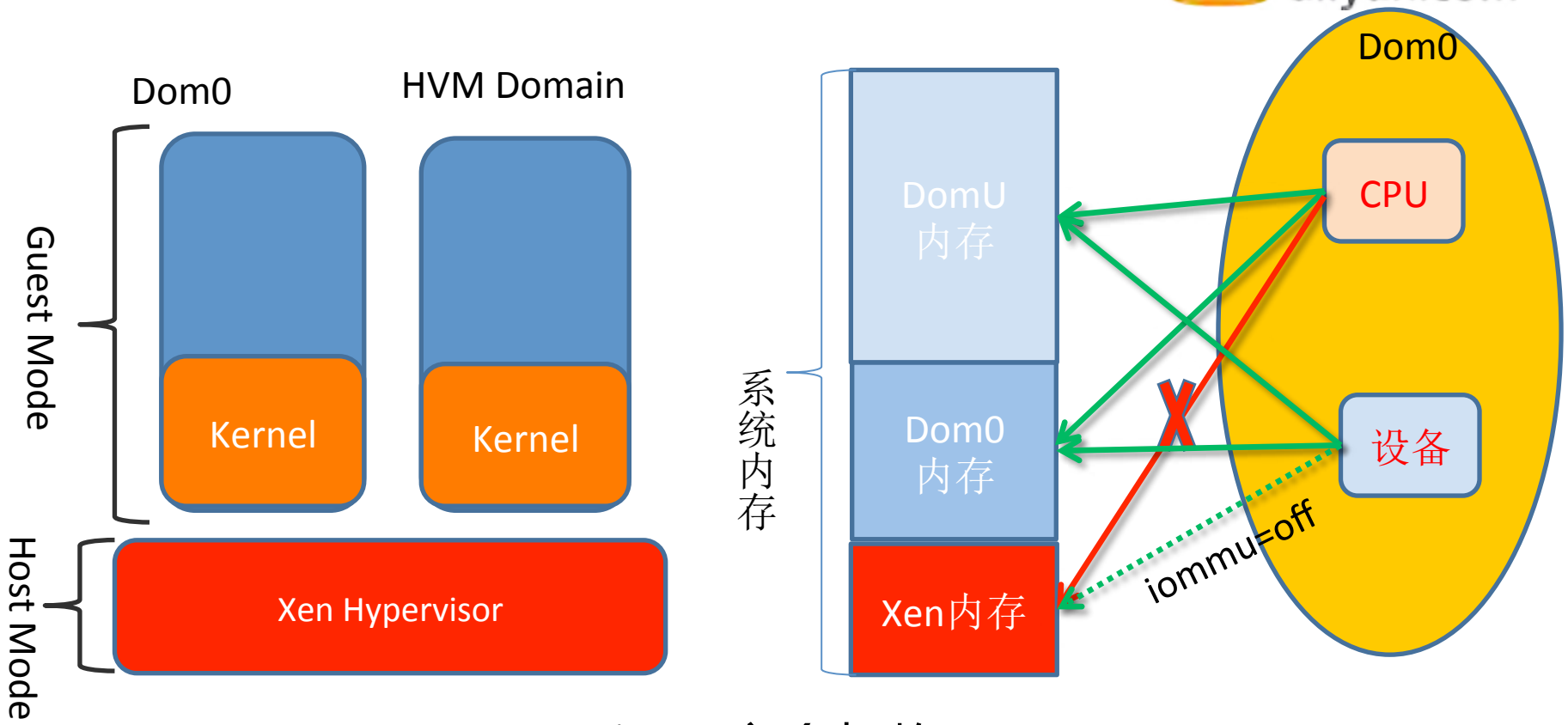
Xen Dom0 内核Hotfix技术

- AliHotfix技术原理
  - 基于函数动态替换技术
  - 新函数会以模块内函数的形式链接入内核
  - 旧函数的第一个指令改成强制跳转指令指向新函数
  - 在替换过程中需要暂停所有CPU，切到一个内核线程并关闭本地中断。
  - 刷新指令缓存，重新让CPU恢复执行
- Hotfix过程中需要注意的点
  - 修复NMI处理函数是不安全的
  - 修复的函数正在内核栈上，修复过程是不安全的
  - 新函数绝对不能调用旧函数，否则无穷递归
  - Inline函数不能被直接修复，需要修复调用者
- Dom0 Hotifx 案例
  - FPU Hotfix，netback hotfix，eflags hotfix
  - ……

Xen Hypervisor Hotfix

- Hypervisor Hotfix需求
  - Xen 安全漏洞: http://xenbits.xen.org/xsa/
  - Xen功能性Bug
- Hypervisor Hotfix挑战极大
  - Xen Hypervisor 逻辑复杂
  - Xen 是type-1 Hypervisor, 不允许Dom0访问Hypervisor内存
  - 线上系统无法新增Hotfix接口
- Hypervisor Hotfix 是史无前例的工作
  - 仅仅是理论上可行的一种方法，无成功先例
  - 如何解决从Dom0 访问 Hypervisor内存
  - 如何精确定位Hypervisor function 物理地址
  - 如何精确替换有问题的代码段和数据段

如何解决Hypervisor 的内存访问？



Xen Hypervisor 安全架构

- Dom0无法通过CPU访问Xen hypervisor内存
- Dom0可通过设备DMA方式访问 Xen hypervisor 内存

如何计算 Hotfix代码/数据的地址？

- Hypervisor load过程
  - Hypervisor首先会被Load到低端地址
  - Hypervisor会把自己Relocate到高端地址
  - Hypervisor 高端地址的计算由系统的E820表决定
- Hypervisor Hotfix 物理地址计算公式
  - 假设需要fix的Hypervisor 函数 地址为0xffff82c480104818 (VA)
  - Hotfix点在Hypervisor内核实际偏移则为 PA’ =VA & 0xffffff
  - 如果E820最后一个内存项为
    - BIOS-e820: pa_start – pa_end (usable)
  - 则要Hotfix函数的物理地址为：
    - PA= pa_start (2M align) + PA’

如何通过DMA访问Hypervisor内存？

- 如何构造DMA请求
  - 不能随意构造不存在的DMA请求
  - 需要截获一个正常DMA请求，修改DMA的目的地址，以及要写入的数据
  - 选取哪个硬件设备， 网卡 ？硬盘？其它？
- 截获DMA请求的方法
  - DMA请求的内存管理来自于两个函数
    - dma_map_sg_attrs/dma_unmap_sg_attrs
  - 利用Alihotfix 替换内核的这两个函数
  - 在新的map_sg/unmap_sg中加入过滤逻辑
  - 筛选出特定的DMA请求，修改DMA目的地址

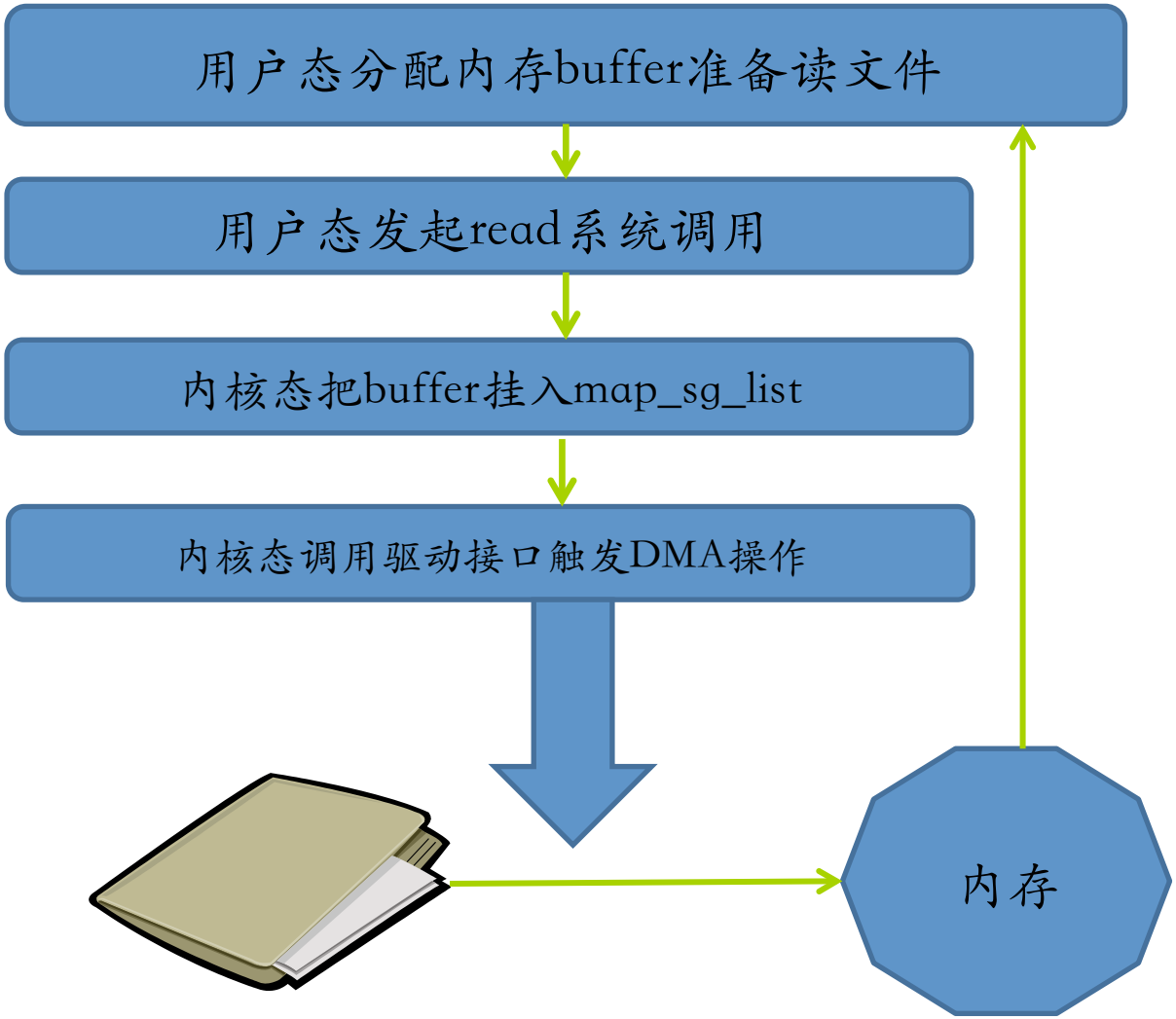
利用硬盘DMA请求Hotfix Hypervisor 内存

Hypervisor Hotfix 方案实现

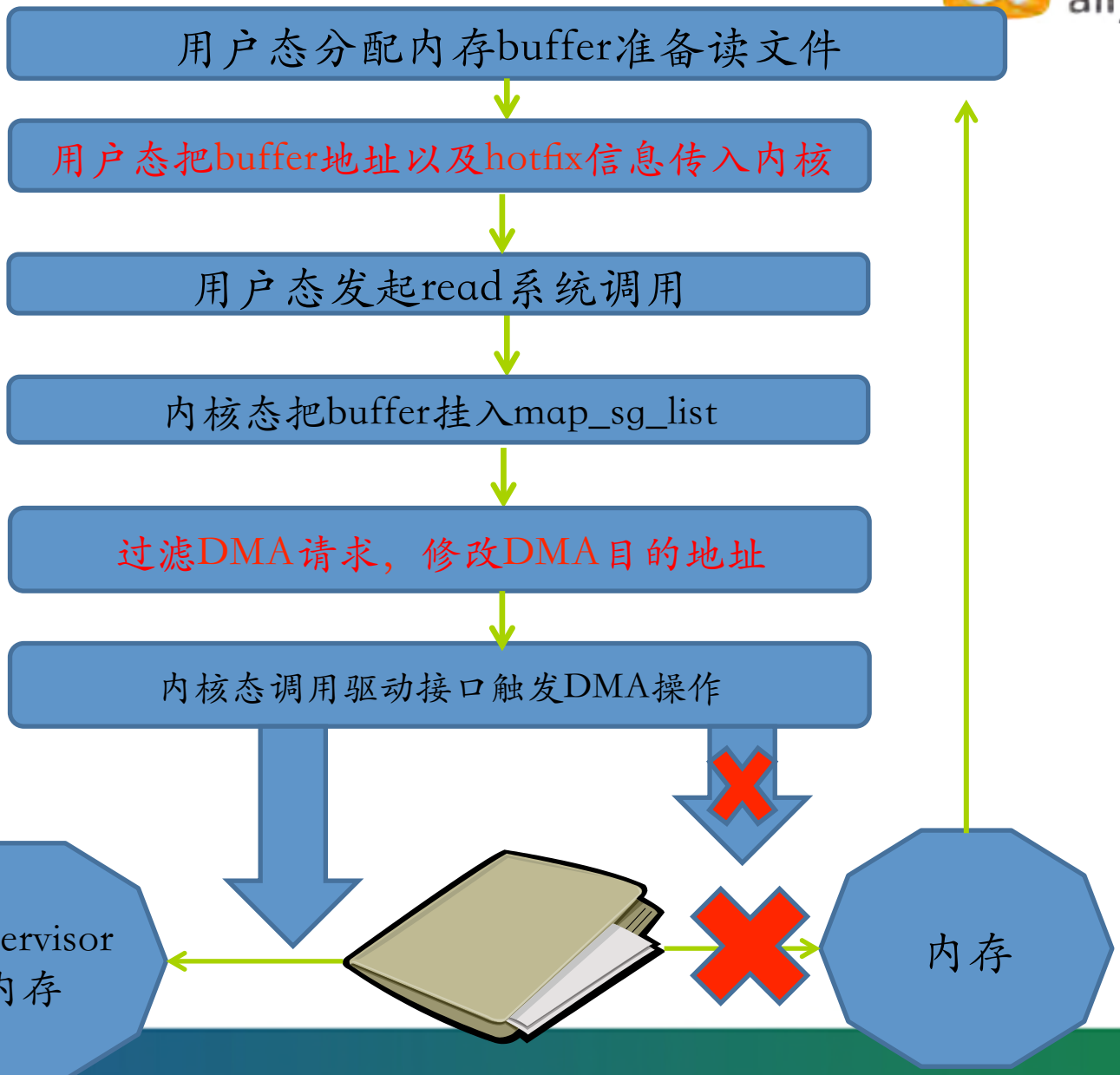
- 内核态层面
  - Hotfix xen_pci_swiotlb_mag_sg/xen_pci_swiotlb_unmap_sg
  - 截获所有的DMA 请求的内存分配操作
- 用户态层面
  - 准备新的二进制代码段或数据段，把需要hotfix的代码和数据存入文件
  - 准备一应用程序，用于读写数据文件触发DMA操作，读数据文件，并提供一buffer
- 用户态和内核态交互
  - 生成一misc device 节点 /dev/hotfix
  - 把需要hotfix的物理地址PA，数据长度，以及用户态 buffer的地址传入内核
  - 内核接受用户态参数后，动态监控是否用户态 buffer加入到sg list中
  - 一旦在sg list中发现用户态传过来的buffer。动态修改dma物理地址和dma 数据长度（来自用户态）
  - 然后继续 DMA操作，通过这种方式把用户态数据文件中准备的数据动态填充到Xen hypervisor内存中

关键路径需要暂停所有VM并做cache flush操作

正常的文件读操作流程


用户态分配内存buffer准备读文件
用户态发起read系统调用
内核态把buffer挂入map_sg_list
内核态调用驱动接口触发DMA操作
内存

Hotfix Hypervisor流程



议程

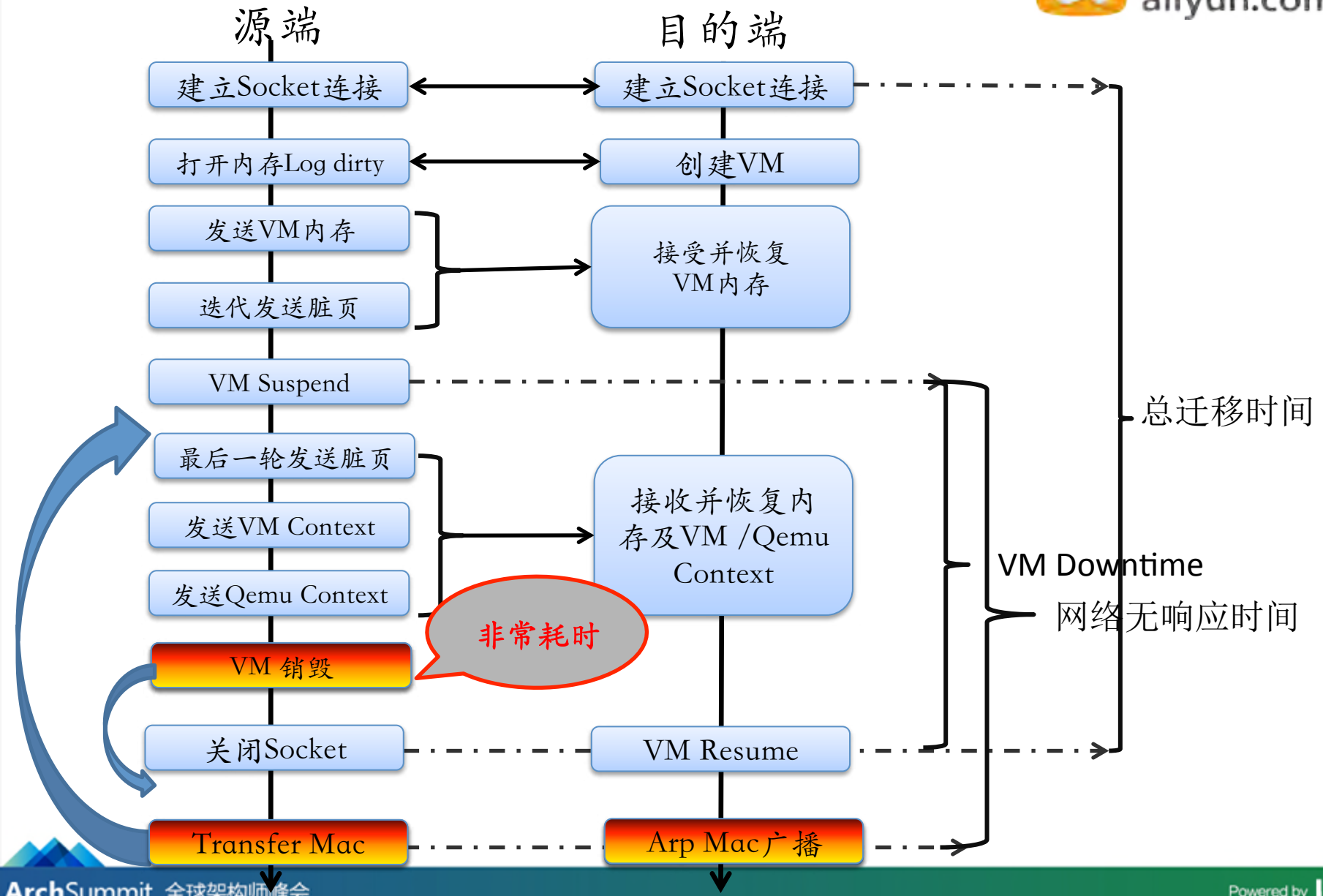
- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

实战案例一

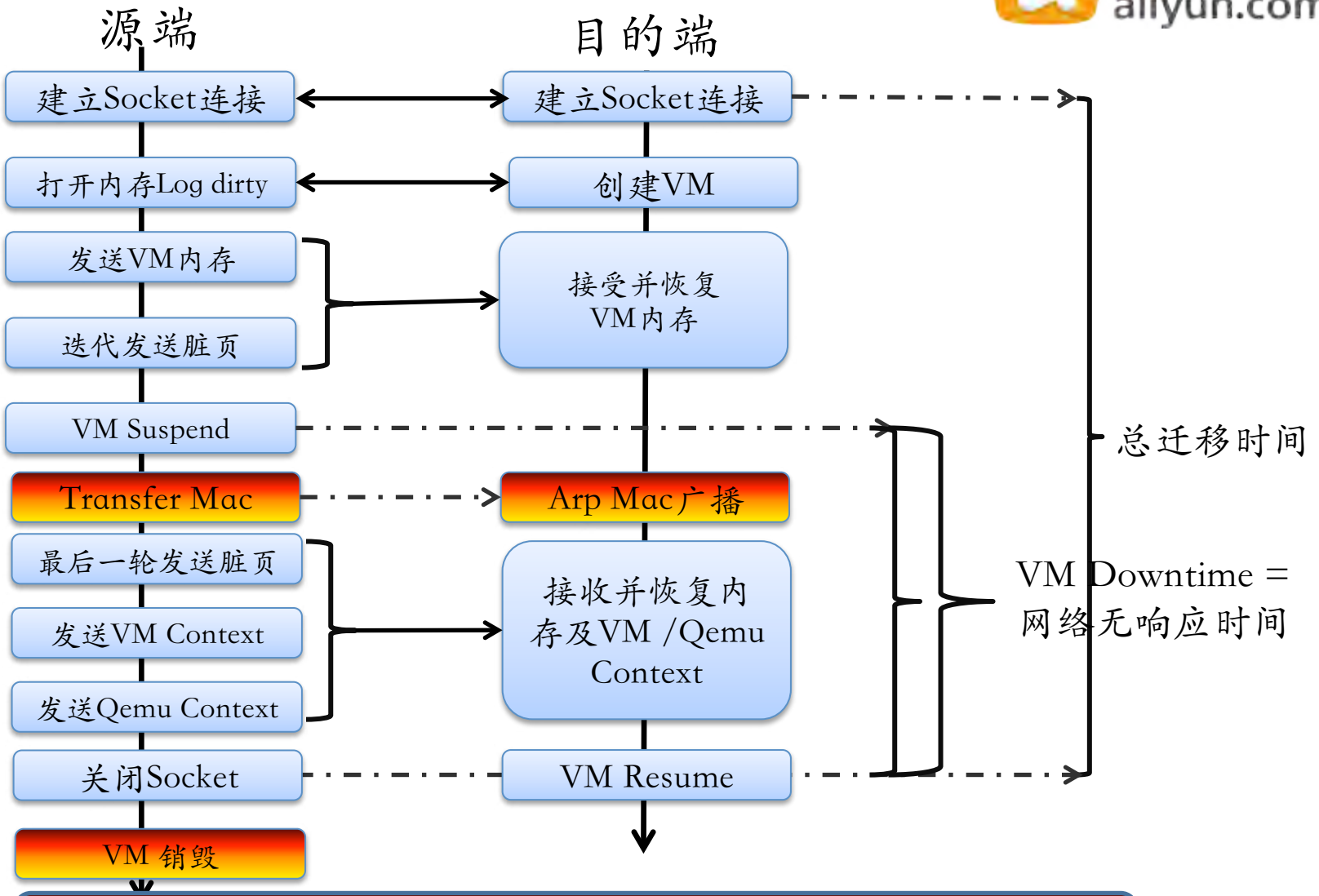
阿里云 aliyun.com

ECS VM 热迁移

优化前的热迁移流程


源端
目的端
建立Socket连接
建立Socket连接
打开内存Log dirty
创建VM
发送VM内存
迭代发送脏页
接受并恢复VM内存
VM Suspend
最后一轮发送脏页
发送VM Context
发送Qemu Context
接收并恢复内存及VM /Qemu Context
非常耗时
VM 销毁
关闭Socket
VM Resume
Transfer Mac
Arp Mac广播
总迁移时间
VM Downtime
网络无响应时间

优化后的VM迁移流程


源端
目的端
建立Socket连接
建立Socket连接
打开内存Log dirty
创建VM
发送VM内存
迭代发送脏页
接受并恢复VM内存
VM Suspend
Transfer Mac
Arp Mac广播
最后一轮发送脏页
发送VM Context
发送Qemu Context
接收并恢复内存及VM /Qemu Context
关闭Socket
VM Resume
VM 销毁
总迁移时间
VM Downtime = 网络无响应时间
VM downtime和网络无响应时间大大降低

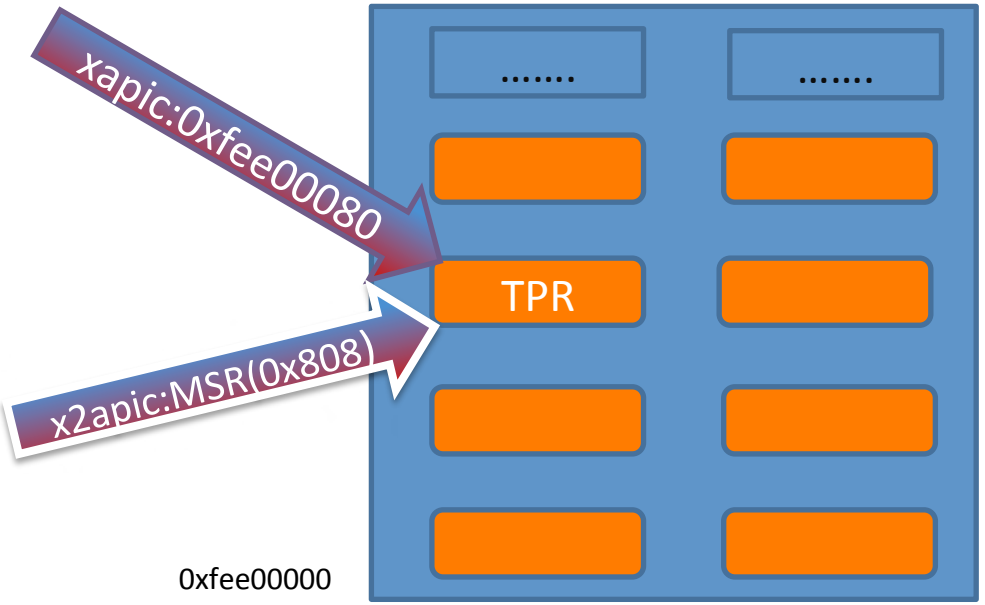
实战案例二

阿里云 aliyun.com

XSA-108事件

问题根源

- KVM 引入了客户机x2apic 支持
  - 增强APIC访问的效率
  - Patch来自KVM maintainer
  - MSR寄存器组的边界计算错误
  - KVM代码进行了出错处理，因此幸免
- Xen 移植了KVM 的Patch
  - Xen无相关的错误处理，造成安全漏洞
  - 每个vCPU就造成4个页面泄露
  - 黑客可以通过重复启动VM，获得几乎所有的hypervisor内存



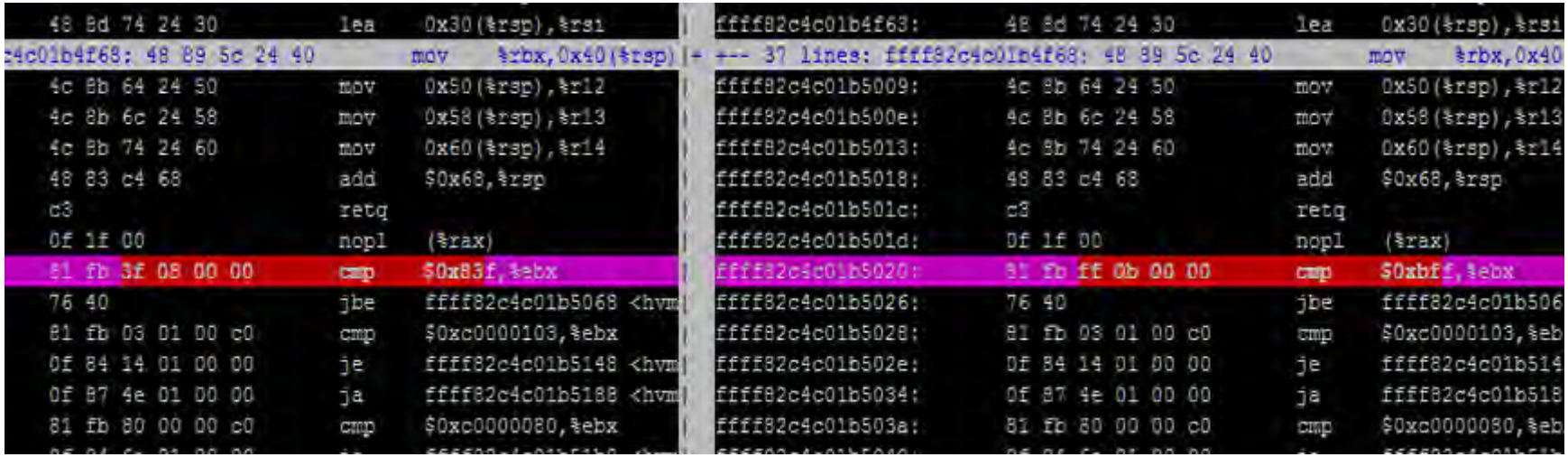
Registers in APIC page

PA=0xfee00000+ (MSR_index -0x800) *0x10

造成了虚拟机逃逸事件

如何修复？

- 方法1：打补丁后重启机器
  - 友商们的方法
- 方法2：Hypervisor Hotfix
  - 阿里云研发的Hypervisor Hotfix方案

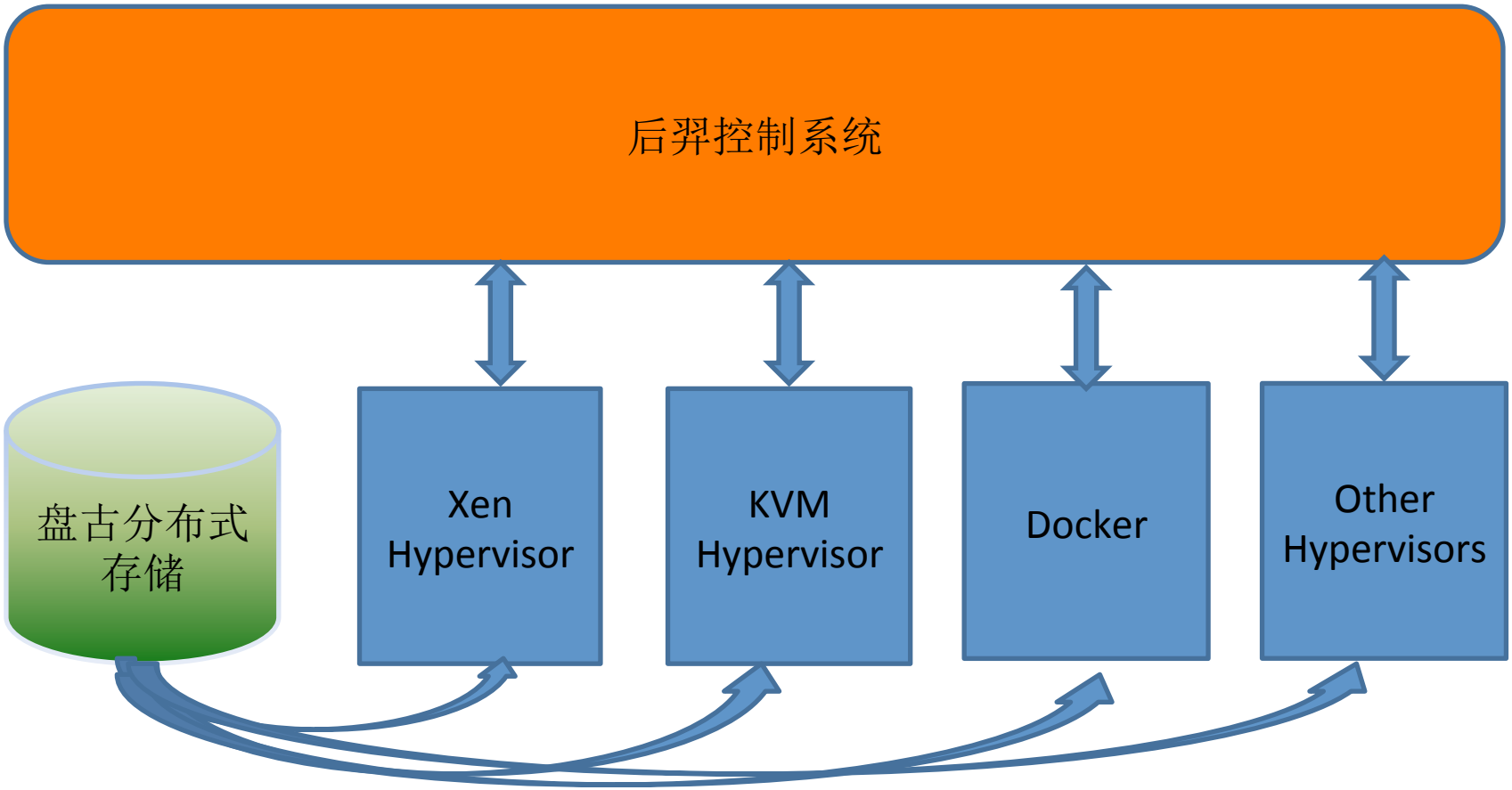


议程

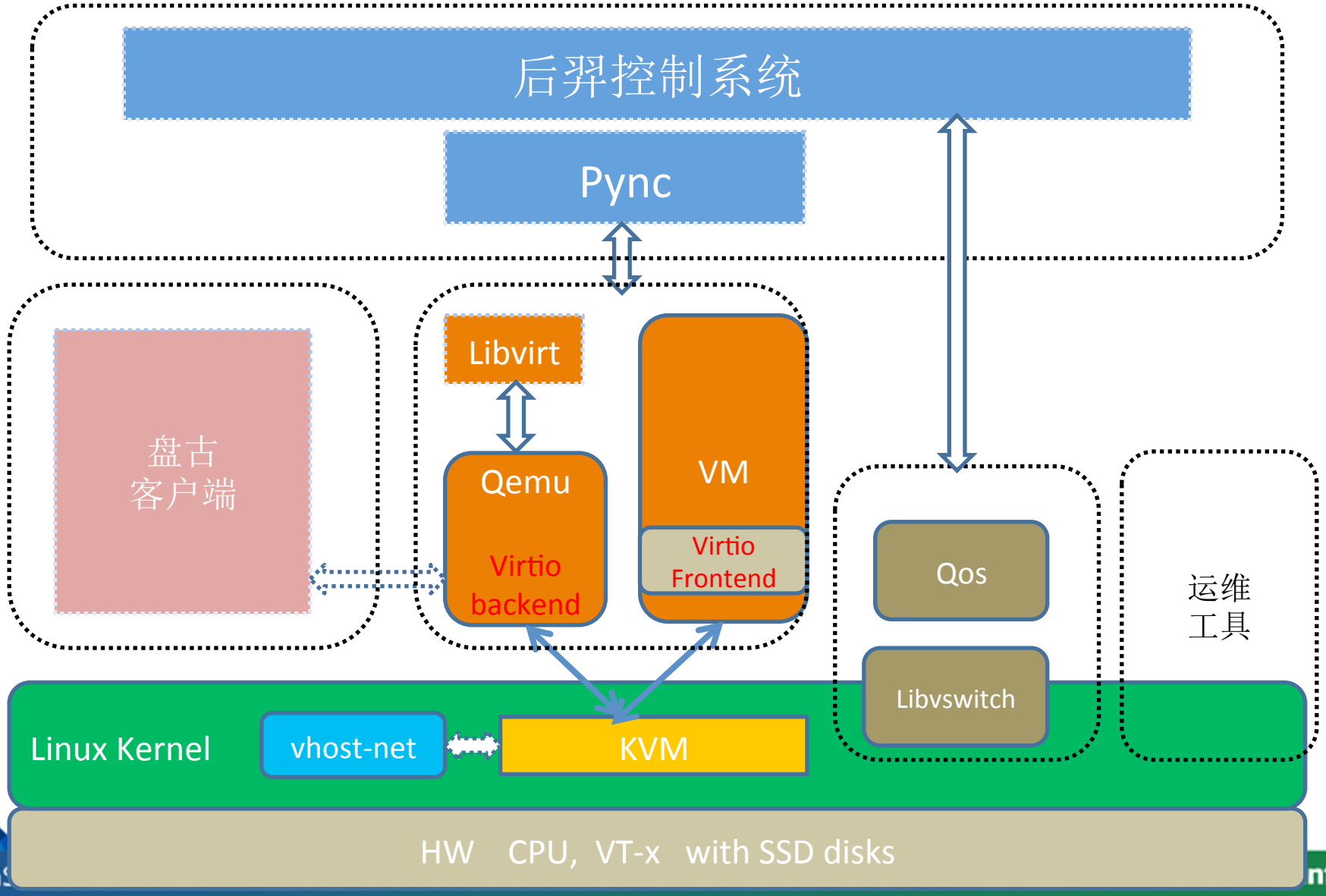
阿里云 aliyun.com

- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

ECS下一代虚拟化架构



基于KVM Hypervisor 架构实现



基于KVM Hypervisor 架构实现

- 设计特点
  - 所有组件都支持热升级，升级过程用户无感知
    - 实现非常具有挑战性
    - KVM Hypervisor，Qemu，vhost-net，前后端驱动，盘古客户端
  - 网络提供基于SRIOV的技术的Qos功能
    - 计算、存储网络流量隔离，消除系统性能抖动
  - 高速热迁移支持
    - VM Downtime降至毫秒级
    - 利用内存实时压缩算法，降低迁移时间
    - 网络无响应时间缩短到1s以内

研发迭代速度快，对Bug和安全漏洞彻底免疫

议程

- 阿里云弹性计算服务ECS介绍
- ECS虚拟化架构及关键技术
  - ECS虚拟化架构
  - 硬件虚拟化技术
  - 虚拟机热迁移技术
  - Hypervisor 热补丁技术
- ECS实战案例分享
- 阿里云ECS下一代虚拟化架构设计
- 未来展望

未来工作展望

阿里云 aliyun.com

- ECS虚拟化核心技术研发方向
  - 持续优化热点迁移技术
  - GPU虚拟化支持
  - LXC/cgroup/Docker支持
  - CPU 热插拔技术
  - 内存热插拔技术
  - VM fork 技术
  - 全部组件的热升级技术
  - 优化NUMA支持，获得更好的系统性能
- 提供更加富有弹性的计算服务
  - 结合ESS服务，提供计算资源的动态伸缩
- 我们一直在奋斗。。。

Q&A

Thanks!

We are Hiring!!!

Email: xiantao.zxt@alibaba-inc.com