



IB00109 云计算技术

授课教师：姜婧妍

jiangjingyan@sztu.edu.cn

2023年





第二章

虚拟化技术

授课教师：姜婧妍

jiangjingyan@sztu.edu.cn

2023年

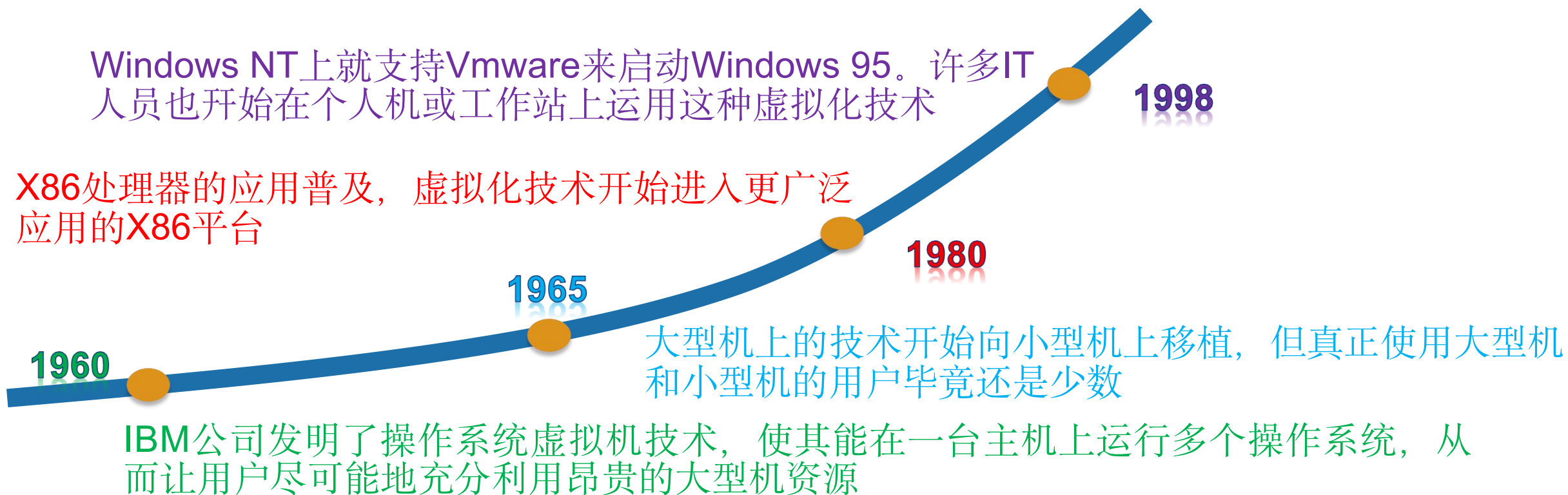


目录 CONTENTS



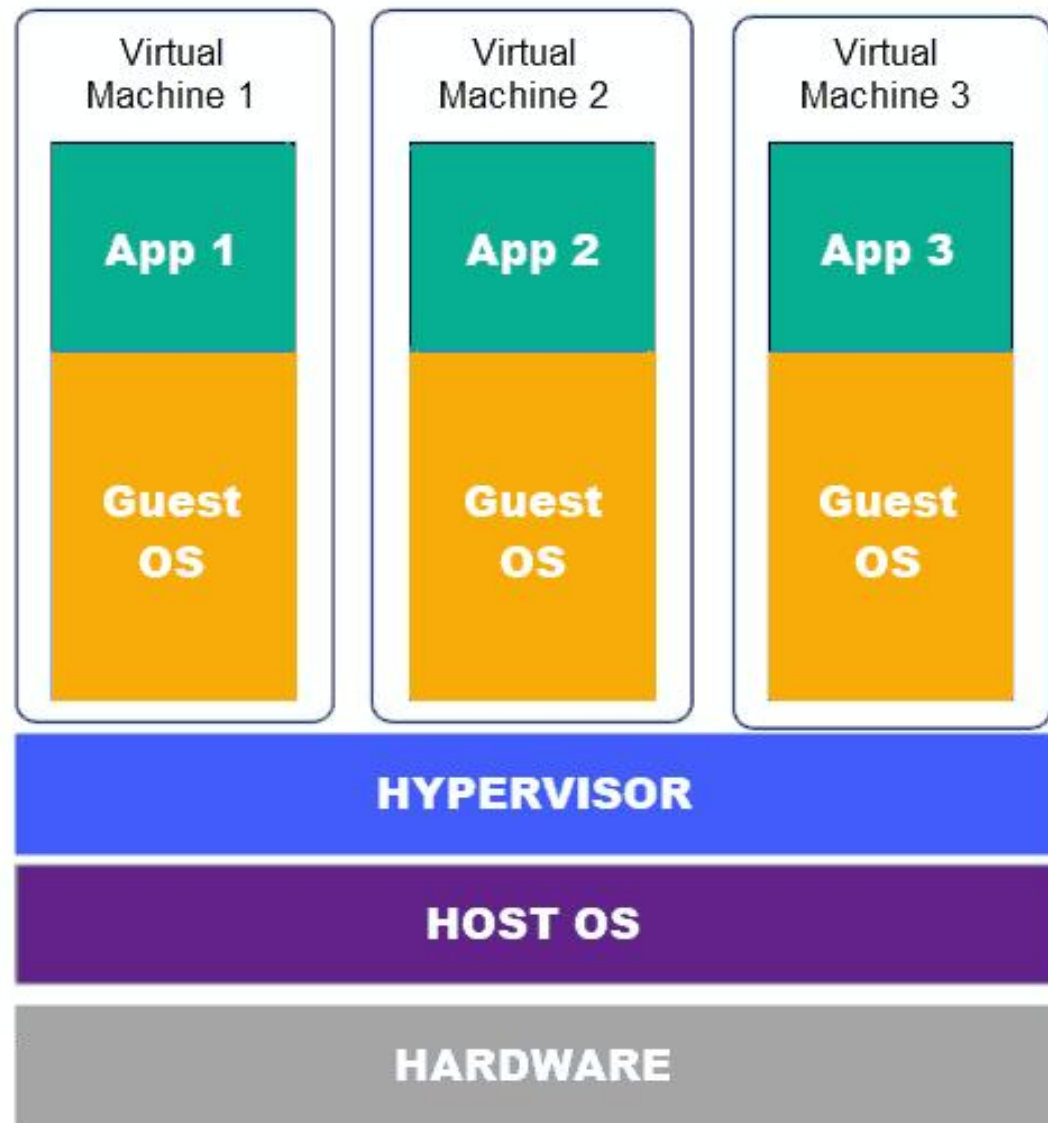
1. 虚拟化简介
2. 计算虚拟化
3. 存储虚拟化
4. 网络虚拟化
5. 桌面虚拟化
6. 容器技术

- **概念：** 虚拟化，是指将一台计算机虚拟为多台逻辑计算机，每个逻辑计算机可相互独立运行而互不影响，从而显著提高计算机的工作效率。
- **历史：**

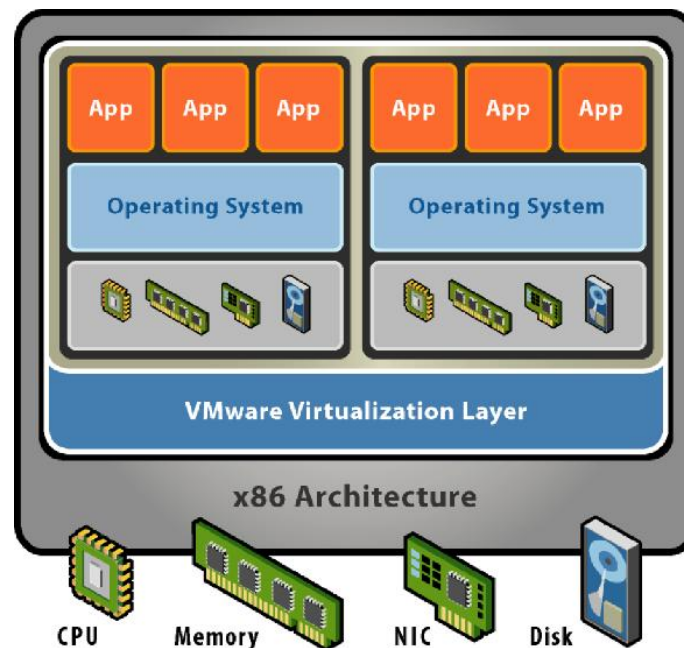
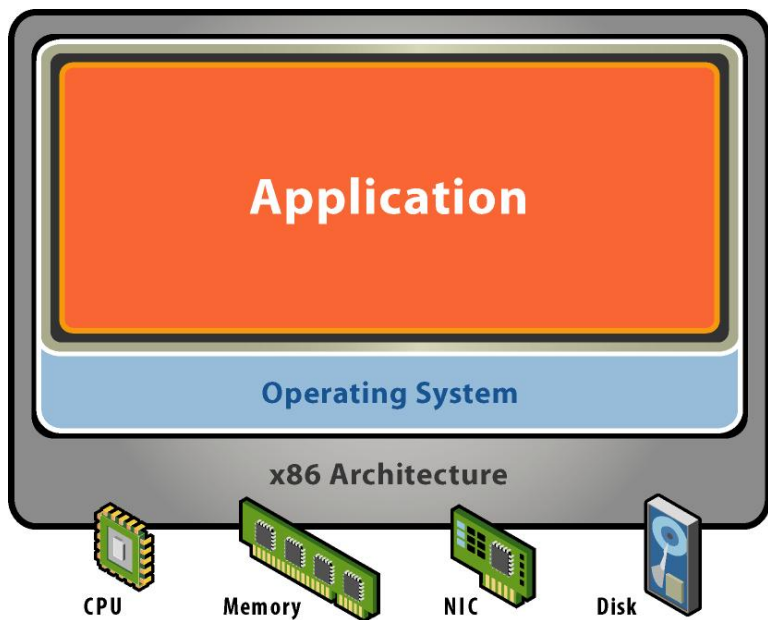


虚拟化中的几个概念

- Host Machine: 物理机
- Host OS: 运行在物理机之上的OS
- Hypervisor : 又称虚拟机监控器
(Virtual Machine Monitor, VMM)
- Virtual Machine: 虚拟出来的虚拟机
- Guest OS: 运行在虚拟机之上的OS



- 虚拟化：是将物理IT资源转换为虚拟IT资源的过程。
- 作用：通过该技术将一台计算机虚拟为多台逻辑计算机。在一台计算机上同时运行多个逻辑计算机，每个逻辑计算机可运行不同的操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。



虚拟化带来的好处

01

提高资源利用率

通过整合服务器可以将共用的基础架构资源聚合到资源池中，打破原有的一台服务器一个应用程序的模式。

02

降低成本，节能减排

通过使用虚拟化，可以使所需的服务器及相关 IT 硬件的数量变少

03

统一管理

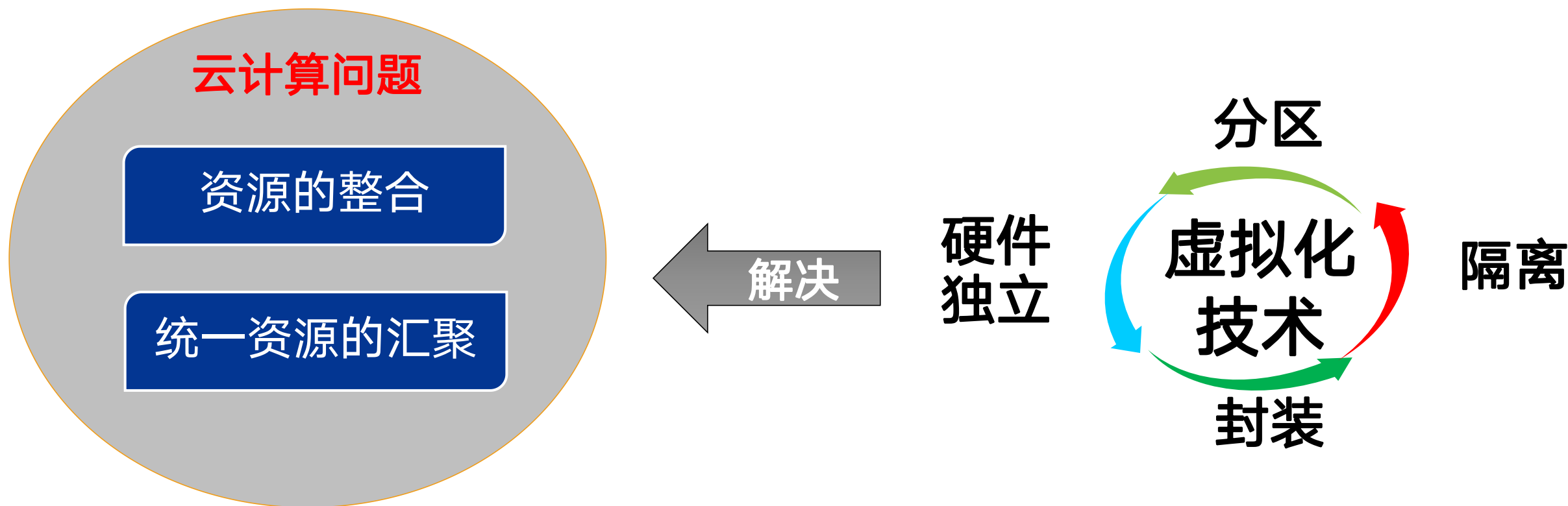
虚拟化系统将资源整合，在管理上十分方便

04

提高安全性

用户可以在一台计算机上模拟出多个不同的操作系统，在虚拟系统下的各个子系统相互独立

和云计算之间的关系：



- 虚拟化技术特点：

分区：大型的、扩展能力强的硬件可被用来作为多台独立的服务器使用；在一个单独的物理系统上，可以运行多个虚拟的操作系统和应用；计算资源可以被放置在资源池中，并能够被有效地控制



- 虚拟化技术特点：

隔离：虚拟化能够提供理想化的物理机，每个虚拟机互相隔离；数据不会在虚拟机之间泄露；应用只能在配置好的网络上进行通讯。



- 虚拟化技术特点：

封装：虚拟单元的所有环境被存放在一个单独文件中；为应用展现的是标准化的虚拟硬件，确保兼容性；整个磁盘分区被存储为一个文件，易于备份、转移和拷贝



- 虚拟化技术特点：

硬件独立：可以在其他服务器上不加修改的运行虚拟机。虚拟技术支持高可用性、动态资源调整，极大地提高系统的可持续运行能力



- 虚拟化技术是一种**思想**：

IT界的所有硬件或者软件都可以一种“服务组合”的抽象思想来处理，即形成一个可被用户灵活调用的资源池，从而实现外部用户业务系统和IT软硬件环境的解耦。

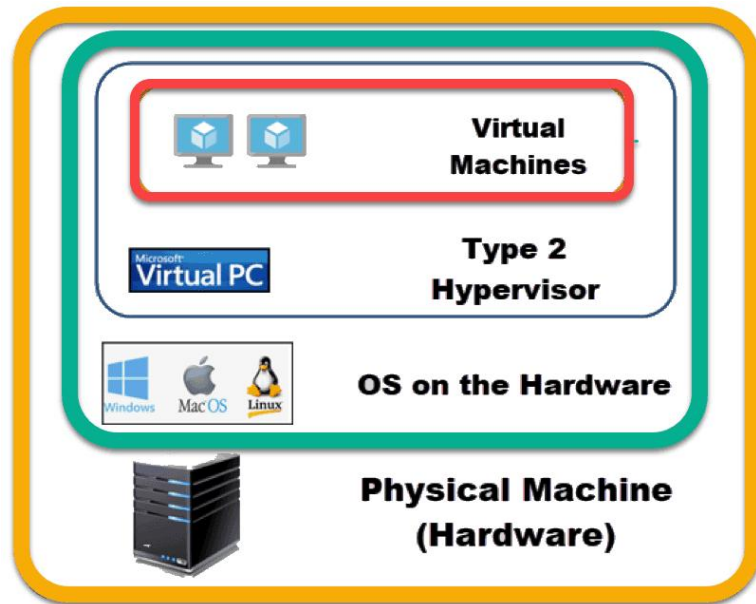
这意味着，外部用户业务系统无须了解软硬件的实现细节，就能方便地使用各式各样的软硬件资源，就好像这些资源放在一个黑箱里一样，只需通过接口就能访问，感受不到其真正的实体和虚体的区别，而这也通常被称为“用户透明化”。

① 从虚拟平台角度划分

- **全虚拟化：** 虚拟的操作系统,与底层的硬件完全隔离,由中间的Hypervisor层转化。典型的代表有Vmware WorkStation, Microsoft Virtrual Server。
- **半虚拟化：** 在虚拟机的操作系统当中加入特定的虚拟化指令,可以直接通过Hypervisor调用硬件资源,免除了Hypervisor层转换指令的开销。典型代表有Xen, Hyper-V。

② 从虚拟化的层次划分

- **软件辅助的虚拟化技术：** 通过软件的方法,让客户机的特权指令陷入异常,从而触发宿主机进行虚拟化。如Hyper-V等。
- **硬件支持的虚拟化技术：** 在X86处理系统架构中加入新的指令机以及运行模式完成虚拟化操作,进而对硬件资源进行直接调用,进行虚拟化。如AMD-V等。

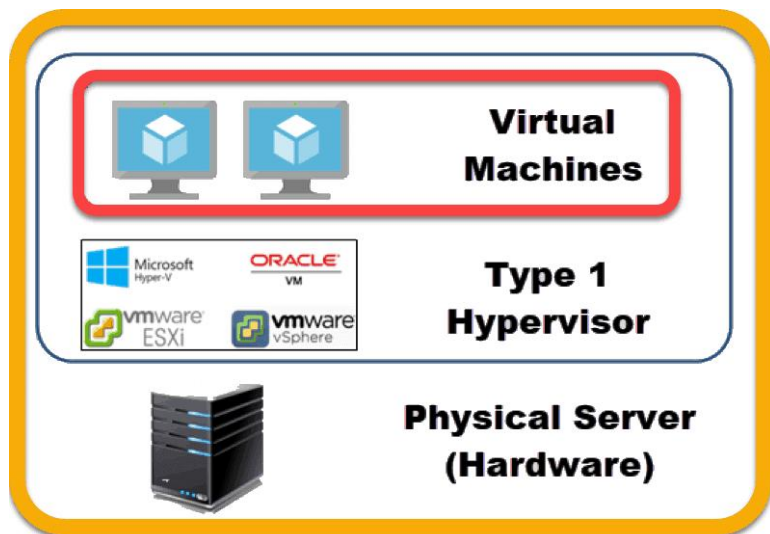


③从虚拟化的实现结构划分

- **基于操作系统的虚拟化：** 在一个已存在的操作系统上安装虚拟化软件
- **特点：**
 - 简单、易于实现
 - 安装和运行虚拟化程序依赖于主机操作系统对设备的支持
 - 有两层OS，管理开销较大，性能损耗大
 - 虚拟机对各种物理设备的调用，都通过虚拟化层和宿主机的OS一起协调才能完成
- **例子：** Vmware Workstation和VirtualBox等

③从虚拟化的实现结构划分

- **基于硬件的虚拟化：**将虚拟化软件直接安装在物理主机硬件上。
- **特点：**
 - 不依赖于主机操作系统
 - 支持多种操作系统，多种应用
 - 依赖虚拟化层进行管理
 - 需要对虚拟层内核进行开发
- 例子：VMvare ESX、Xen等



④从虚拟化在云计算的应用领域进行划分

- **服务器虚拟化：**将一台服务器虚拟成多台服务器进行使用
- **存储虚拟化：**将整个云系统的存储资源进行统一整合管理
- **应用程序虚拟化：**把应用程序对底层硬件和系统的依赖抽取出来，从而解耦
- **平台虚拟化：**集成各种开发资源虚拟出一个面向开发人员的统一接口，如监控视频平台、消息平台、短信平台
- **桌面虚拟化：**将用户的桌面以使用的终端进行分离，进行解耦



① 完全虚拟化

- 最流行的虚拟化方法使用名为hypervisor的一种软件,在虚拟服务器和底层硬件之间建立一个抽象层。VMware和微软的VirtualPC是代表该方法的两个商用产品,而基于核心的虚拟机(KVM)是面向Linux系统的开源产品。
- hypervisor可以捕获CPU指令,为指令访问硬件控制器和外设充当中介。因而,完全虚拟化技术几乎能让任何一款操作系统不用改动就能安装到虚拟服务器上,而它们不知道自己运行在虚拟化环境下。主要缺点是, hypervisor给处理器带来开销。



② 准虚拟化

- 完全虚拟化是处理器密集型技术,因为它要求hypervisor管理各个虚拟服务器,并让它们彼此独立。减轻这种负担的一种方法就是,改动客户操作系统,让它以为自己运行在虚拟环境下,能够与hypervisor协同工作。这种方法就叫准虚拟化(para-virtualization)。
- Xen是开源准虚拟化技术的一个例子。操作系统作为虚拟服务器在Xen hypervisor上运行之前,它必须在核心层面进行某些改变。因此, Xen适用于BSDLinux, Solaris及其他开源操作系统,但不适合对像Windows这些专有的操作系统进行虚拟化处理,因为它们无法改动。
- 准虚拟化技术的优点是性能高。

**ubuntu/images-testing/hvm-ssd/ubuntu-cosmic-daily-amd64-server-20190318** - ami-0033f98233eadd8fd

Canonical, Ubuntu, 18.10, UNSUPPORTED daily amd64 cosmic image build on 2019-03-18

Root device type: ebs

Virtualization type: hvm

ENA Enabled: Yes

[Select](#)

64-bit (x86)

**ubuntu/images-testing/ebs-ssd/ubuntu-trusty-daily-amd64-server-20190424** - ami-00365845c96ac739f

Canonical, Ubuntu, 14.04 LTS, UNSUPPORTED daily amd64 trusty image build on 2019-04-24

Root device type: ebs

Virtualization type: paravirtual

ENA Enabled: No

[Select](#)

64-bit (x86)

**Windows_Server-2019-English-Full-Base-2021.09.15** - ami-0428fc1ee1bde045a

Microsoft Windows Server 2019 with Desktop Experience Locale English AMI provided by Amazon

Root device type: ebs

Virtualization type: hvm

ENA Enabled: Yes

[Select](#)

64-bit (x86)

**Windows_Server-2019-English-Full-ContainersLatest-2021.09.15** - ami-0e8900c6c58aa0b89

Microsoft Windows Server 2019 with Containers Locale English AMI provided by Amazon

Root device type: ebs

Virtualization type: hvm

ENA Enabled: Yes

[Select](#)

64-bit (x86)

**Windows_Server-2019-English-Full-SQL_Server_2017_Standard-2021.09.15** - ami-02698d804aed02860

Microsoft Windows Server 2019 Full Locale English with SQL Standard 2017 AMI provided by Amazon

Root device type: ebs

Virtualization type: hvm

ENA Enabled: Yes

[Select](#)

64-bit (x86)

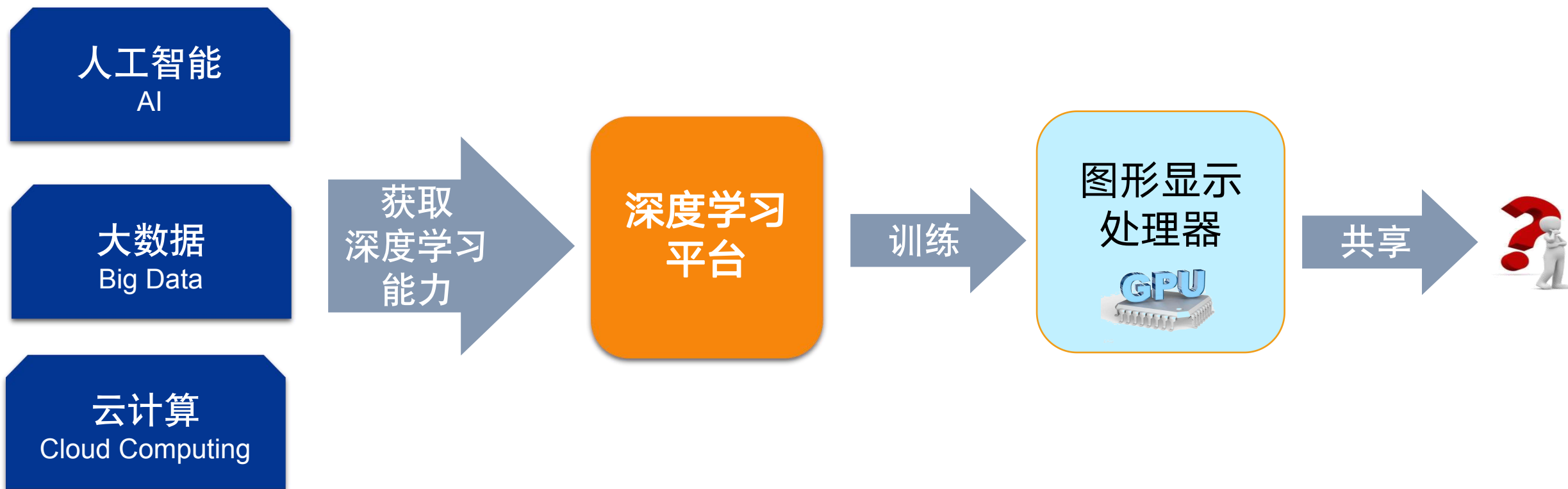
02

计算虚拟化



计算虚拟化

- 场景：公有云计算的深度学习平台



- 定义:

计算虚拟化是指在物理服务器的宿主机操作系统(Host OS)中加入一个虚拟化层(Hypervisor), 在虚拟化层之上可以运行多个客户端操作系统(Guest OS)。资源利用率和灵活性。

- 开源计算虚拟化软件: KVM (Kernel-based Virtual Machine)

- CPU虚拟化
- 内存虚拟化
- GPU虚拟化

③ CPU虚拟化技术

- CPU的虚拟化技术可以单CPU模拟多CPU并行，允许一个平台同时运行多个操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。

计算虚拟化 – CPU虚拟化

3.2 CPU全虚拟化

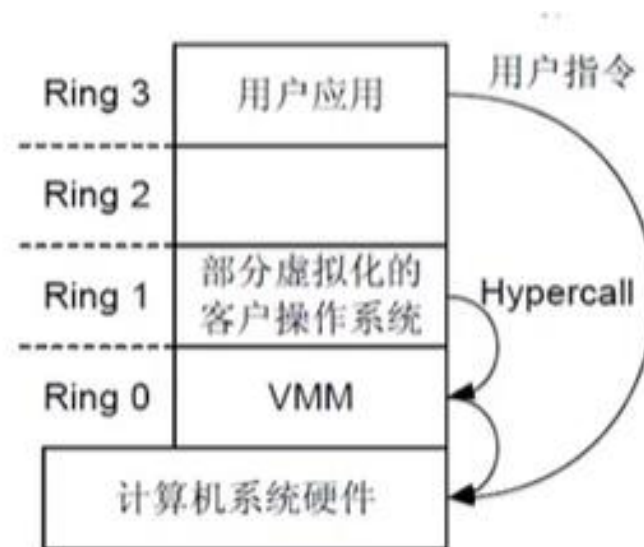
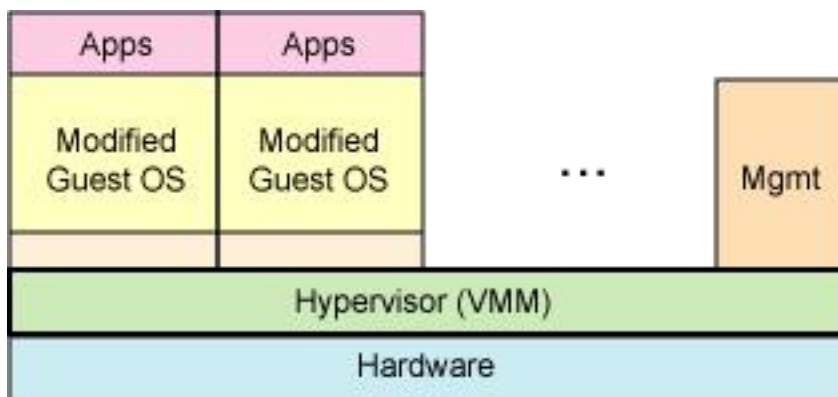
- 全虚拟化指的是虚拟机完完全全的模拟了计算机的底层硬件，包括处理器，物理内存，时钟，各类外设等等。这样呢，就不需要对原有硬件和操作系统进行改动。
- 在虚拟机软件访问计算机的物理硬件就可以看做软件访问了一个特定的接口。这个接口是由VMM（由Hypervisor技术提供）提供的既VMM提供完全模拟计算机底层硬件环境，并且这时在计算机（宿主机）上运行的操作系统（非虚拟机上运行的操作系统）会被降级运行（Ring0变化到Ring1）。



计算虚拟化 - CPU虚拟化

3.2 CPU半虚拟化

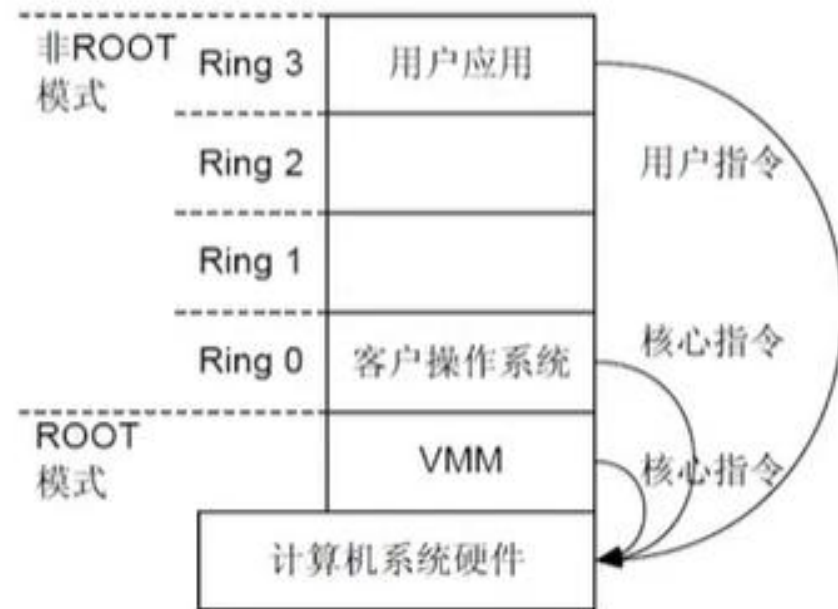
- 主要采用Hypercall技术。Guest OS的部分代码被改变，从而使Guest OS会将和特权指令相关的操作都转换为发给VMM的Hypercall (超级调用)，由VMM继续进行处理。而Hypercall支持的批处理和异步这两种优化方式，使得通过Hypercall能得到近似于物理机的速度。



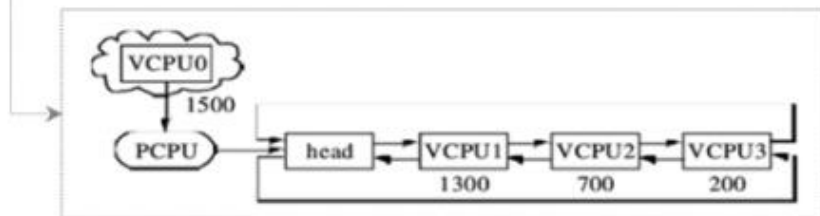
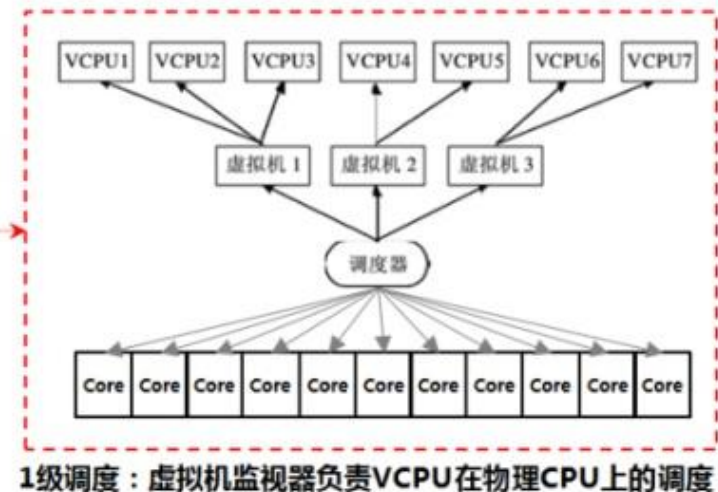
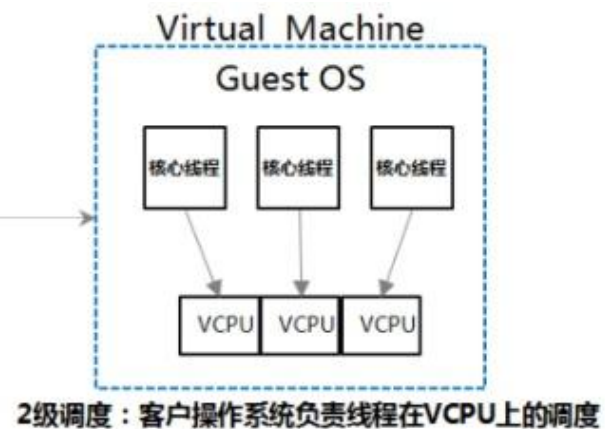
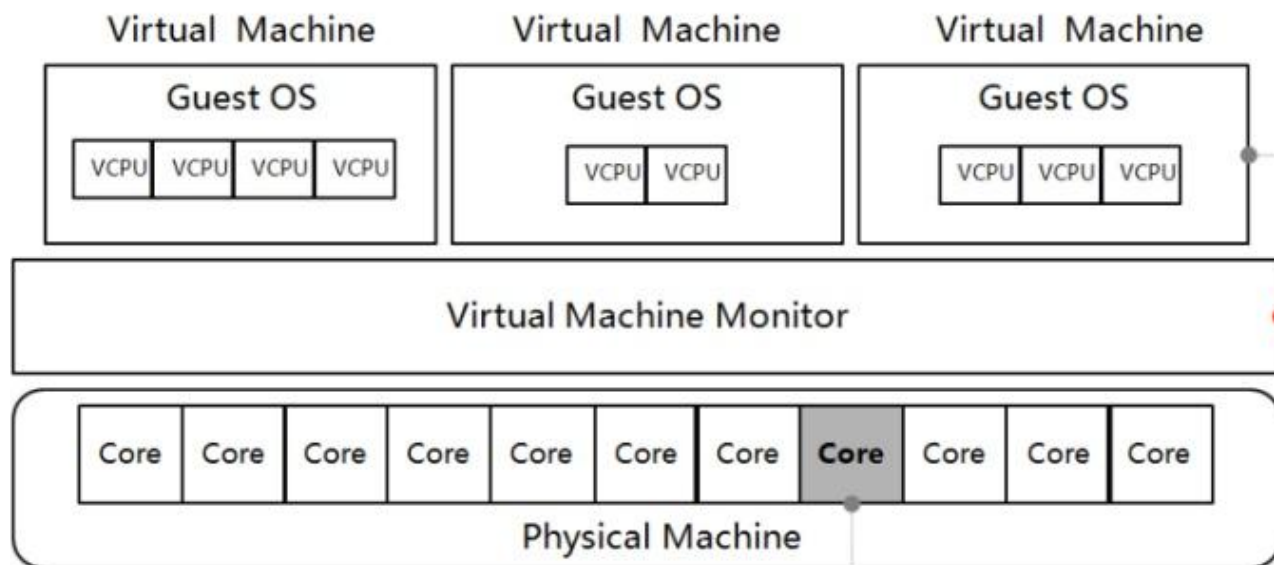
计算虚拟化 - CPU虚拟化

3.3 CPU硬件辅助虚拟化技术

- 目前主要有Intel的VT-x和AMD的AMD-V这两种技术。其核心思想都是通过引入新的指令和运行模式,使VMM和Guest OS分别运行在不同模式(ROOT模式和非ROOT模式)下,且Guest OS运行在Ring 0下。通常情况下, Guest OS的核心指令可以直接下达到计算机系统硬件执行,而不需要经过VMM,当Guest OS执行到特殊指令的时候,系统会切换到VMM,让VMM来处理特殊指令。



计算虚拟化 - CPU虚拟化



物理CPU：每个物理CPU关联着一个VCPU执行队列

计算虚拟化 - 内存虚拟化

- 定义:

通过内存虚拟化来对物理系统内存进行共享，并将其动态分配给虚拟机，操作系统保持着虚拟页到物理页的映射

- 相关概念：物理页和虚拟页

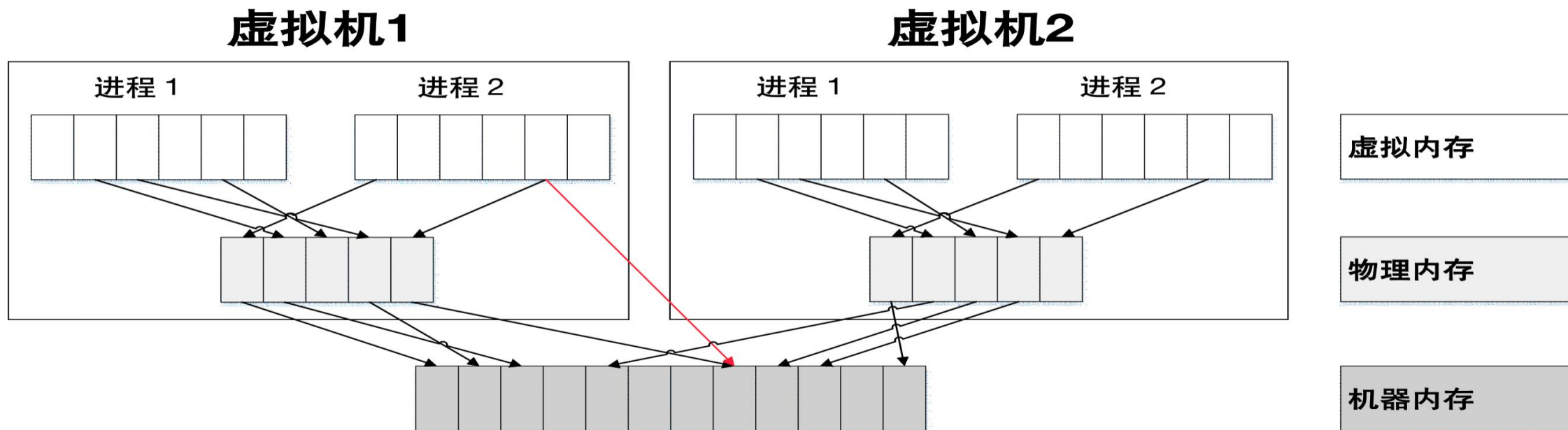
为便于管理，物理内存被分页，就像一本书里面的好多页纸，每张纸上记录了不同的信息。对于32位的CPU来说，每个物理页大小是4K。与之对应的，虚拟页指的是虚拟内存中的分页。

- 内存虚拟化:

让客户机使用一个隔离的、从零开始且具有连续的内存空间，KVM 引入一层新的地址空间，只是宿主主机虚拟地址空间在客户机地址空间的一个映射。

计算虚拟化 - 内存虚拟化

- 图示：KVM 为了运行多台虚拟机于一台物理机器上，需要实现虚拟内存到物理内存到 机器内存直接的地址转换。



计算虚拟化 – GPU虚拟化

- 定义：

通过GPU虚拟化来对GPU资源进行共享。

- GPU虚拟化：

- 虚拟显卡
- 显卡直通
- 显卡虚拟化

计算虚拟化 – GPU虚拟化

- 虚拟显卡

当前主流的虚拟化系统往往采用虚拟显卡来对图像进行处理。

目前虚拟显卡的技术包括：

- 虚拟网络计算机
- Xen 虚拟帧缓存
- VMware 虚拟图形显示处理器
- 独立于虚拟机管理器的图形加速系统

计算虚拟化 – GPU虚拟化

- 显卡直通

显卡直通指绕过虚拟机管理系统，将GPU 单独分配给某一虚拟机，只有该虚拟机拥有使用GPU的权限。

这种方法保留了GPU的完整性和独立性，可以达到与非虚拟化情况下相似的性能，且可以用来进行通用计算。但是显卡直通需要利用显卡的一些特性，并且仅有部分GPU设备可以使用，兼容性差

• 显卡虚拟化

将显卡使用时间进行分片，一般可以根据需求切分成不同的大小的时间分片，将这些分片分配给虚拟机进行使用。其实现原理其实就是利用应用层接口虚拟化，利用应用程序编程接口 进行重定向，在应用层进行拦截与GPU相关的API接口，通过重定向（仍使用GPU）的方式完成相应功能，再将执行结果返回到相应的应用程序。

场景回顾：

针对本小节开始提到的在公有云的深度学习平台案例，可以采用了基于GPU虚拟化的托管集群，实现GPU资源按需使用，从而为深度学习的各种事务提供虚拟化的深度学习能力。

02

存储虚拟化

医疗集团

- 场景：某医疗集团的存储需求



下属多家医院



几百TB的医疗记录



挑战：
海量医疗数据的
高读写速度
高可用性和可靠性

- 定义:

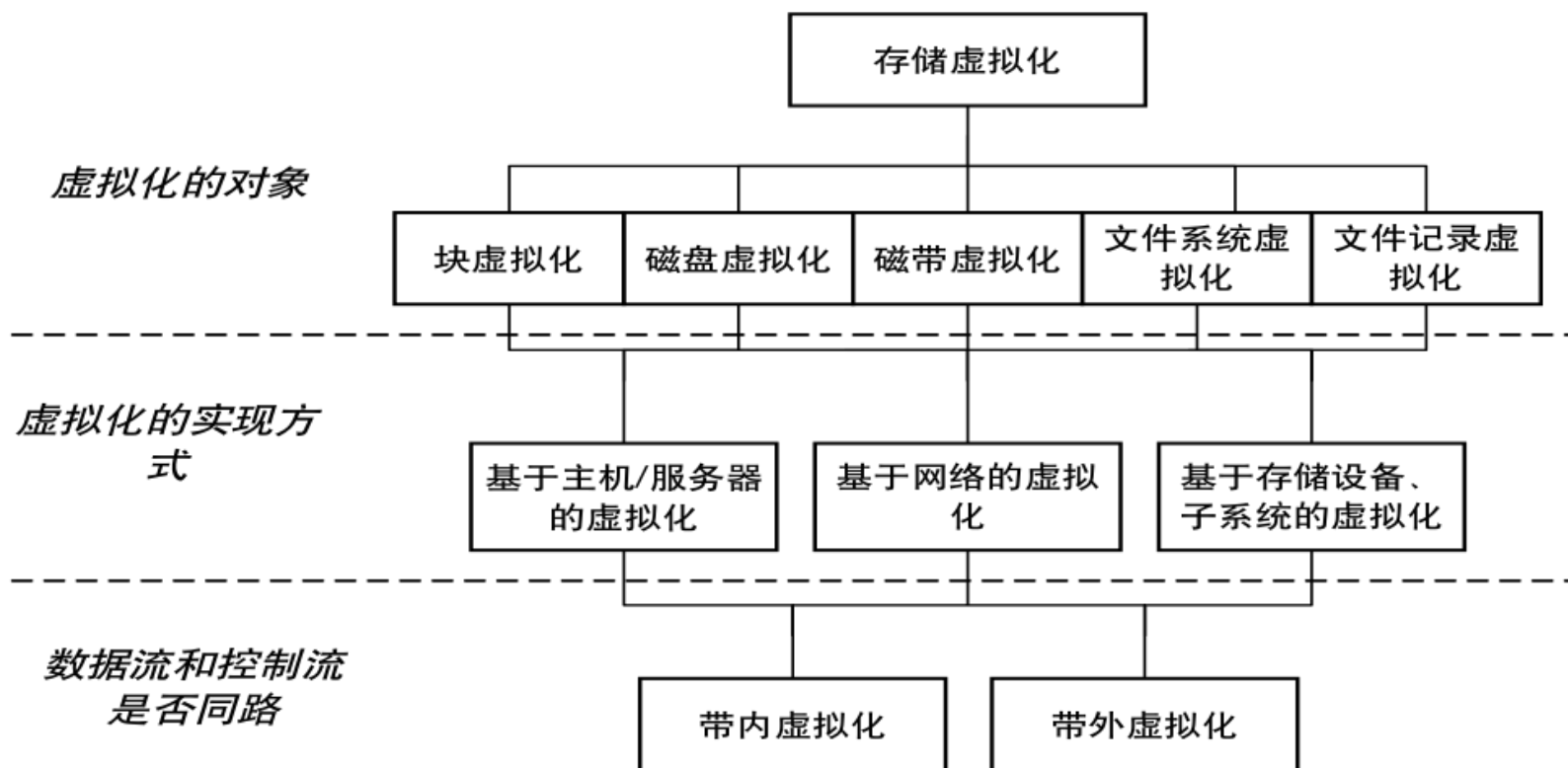
对硬件存储资源进行抽象化，通过对存储系统或存储服务内部的功能进行隐藏、隔离及抽象，使存储与网络、应用等管理分离，存储资源得以合并，从而提升资源利用率。

- 分类:

根据SNIA(Storage Networking Industry Association)的分类方法，可将存储虚拟化技术从不同角度进行分类。

• 分类:

根据SNIA(Storage Networking Industry Association)的分类方法, 可将存储虚拟化技术从不同角度进行分类



- 挑战：
 - 低成本
 - 易实现
 - 灵活性
 - 可扩展性

基于主机

利用安装在一个或多个主机上的控制软件来实现存储虚拟化的控制和管理：

- **最易于实现且其设备成本最低**，因为不需要增加任何附加的硬件
- **扩展性较差**，因为由于运行在主机上的控制软件会占用主机的处理时间。
- **灵活性较差**，因为不同存储厂商软硬件的差异而带来不必要的互操作性开销。

基于存储设备

利用可以提供相关功能的存储模块来进行虚拟化：

- ❑ **可扩展性差**，如果没有使用第三方的虚拟软件，基于存储设备的虚拟化往往只能提供不完全的存储虚拟化解决方案。
- ❑ **灵活性差**，因为使用这种方法来提供虚拟化意味着最终将锁定某一家特定的存储供应商。
- ❑ **较容易实现**，因为容易与某个特定存储供应商的设备相协调。

基于网络

(1)基于互联设备的存储虚拟化

基于互联设备的虚拟化方法能够运行于使用标准操作系统的专用服务器上。该方法相对于基于主机的虚拟化方法具有**易使用、设备低廉**等优势。

(2)基于路由器的存储虚拟化

基于路由器的方法是利用路由器的固件来实现存储虚拟化的功能。

由于路由器可以服务于每一台主机，且大多数控制模块存在于这些路由器中，所以该方式相对于基于主机和大多数基于互联设备的方法具有**更好的性能和效果**。

(3)基于交换机的存储虚拟化

交换机的虚拟化是通过在交换机中嵌入虚拟化模块来实现的

由于在交换机中集成有虚拟化功能，交换机易成为系统的瓶颈，并可能产生**单点故障问题**。

场景回顾：

针对本小节开始提到的医疗集团的存储需求案例，该公司对现有存储设备的虚拟化管理，在此基础上进行数据分层，将整个存储分为两个层次：

- 第一层存储：需要快速访问的患者数据可以放置在高性能的基于光纤的存储系统上；
- 第二层存储：紧急性稍低的数据则可放置在普通的存储系统上。

在多层存储上采用虚拟化技术使存储系统能够进行有效的成本管理，同时还降低了风险、改善了服务，大大节约了成本。

02

网络虚拟化

• 场景：大型网络公司跨地域数据中心的网络管理



数十万服务器分布在全球10多个城市、数十个数据中心，产生的跨城数据中心间流量达到数百G规模

挑战：
数据中心间网络链路带宽资源有限，难以满足业务临时性的大容量传输需求，通常需要业务部门自行搭建VPN平台并通过公网传输，**响应速度慢**。

• 定义：

对物理网络及其组件（比如交换机、端口以及路由器）进行抽象，并从中分离网络业务流量的一种方式。

采用网络虚拟化可以将多个物理网络抽象为一个虚拟网络，或者将一个物理网络分割为多个逻辑网络。

• 举例：

虚拟局域网VLAN（Virtual Local Area Network）：虚拟局域网是一组逻辑上的设备和用户，这些设备和用户并不受物理位置的限制，相互之间的通信就好像它们在同一个网段中一样。

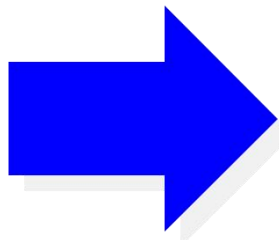
- 网络功能虚拟化：
 - 网络虚拟化通过网络来创建通道、同时利用每一条传输流进行服务。
 - 网络功能虚拟化旨在对网络通信互联模型的四到七层功能进行虚拟化处理。
 - 网络功能虚拟化可以帮助人们为虚拟机或者传输流创建一套服务配置方案，并在网络上建立起抽象结构，最终在特定逻辑环境下构建起虚拟服务。

网络虚拟化

- 基于软件定义网络的网络虚拟化：
- 定义：软件定义网络（Software Defined Network, SDN），是一种新型网络创新架构。
软件定义网络可对网络进行编程，从而构建新的连接方式，而不是像以前那样通过设备来建立网络连接。

- 优点：

传统的网络虚拟化部署需要手动逐跳部署，其效率低下，人力成本很高。



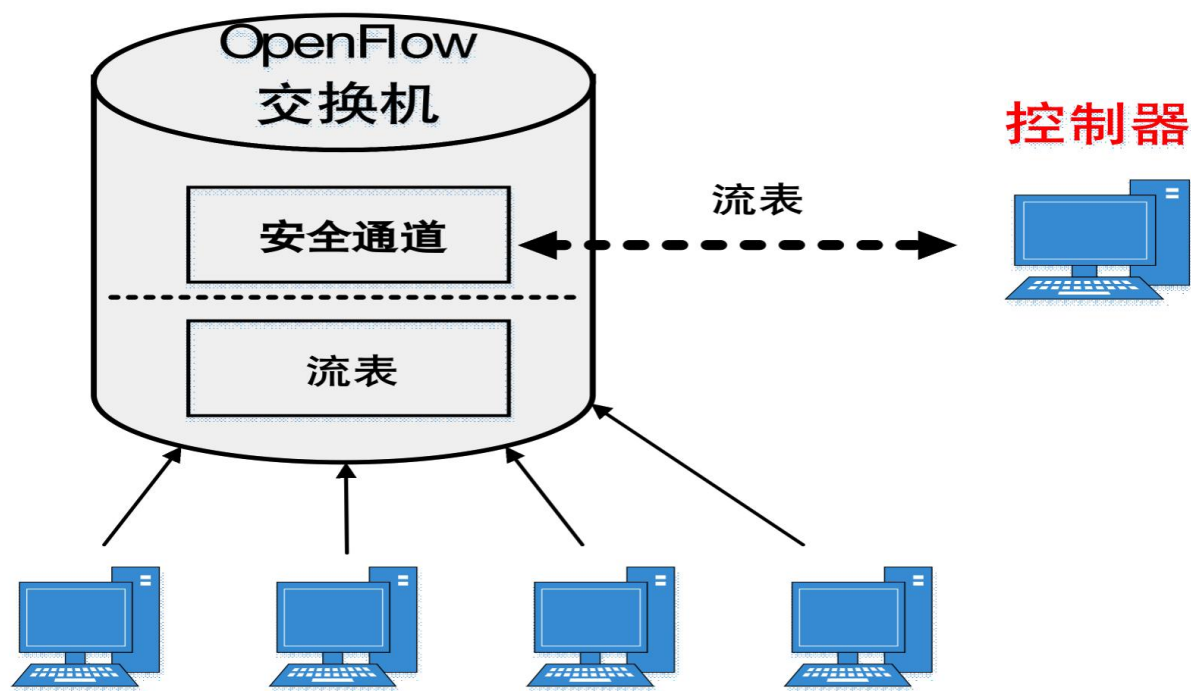
通过软件定义网络，网络可以通过一种编排好的方式进行分配，从而自动化可以快速分配网络服务。

- 软件定义网络的商业应用OpenFlow:
- 定义： OpenFlow 是软件定义网络（SDN，Software Defined Network）的一种协议实现，起源于斯坦福大学，出发点是用于网络研究人员实验其创新网络架构，考虑到实际的网络创新思想需要在实际网络上才能更好地验证，而研究人员又无法修改在网的网络设备，故而提出了 OpenFlow 的控制转发分离架构，将控制逻辑从网络设备中独立出来以便于研究。
- 特点： OpenFlow尽管不是专门为网络虚拟化而生，但是它带来的标准化和灵活性却给网络虚拟化的发展带来无限可能。基于 OpenFlow的软件定义网络，可实现控制层和转发层分离，极大地提升网络的交换速度，满足云计算种高速数据交换和传输的要求。

- 软件定义网络的商业应用OpenFlow:

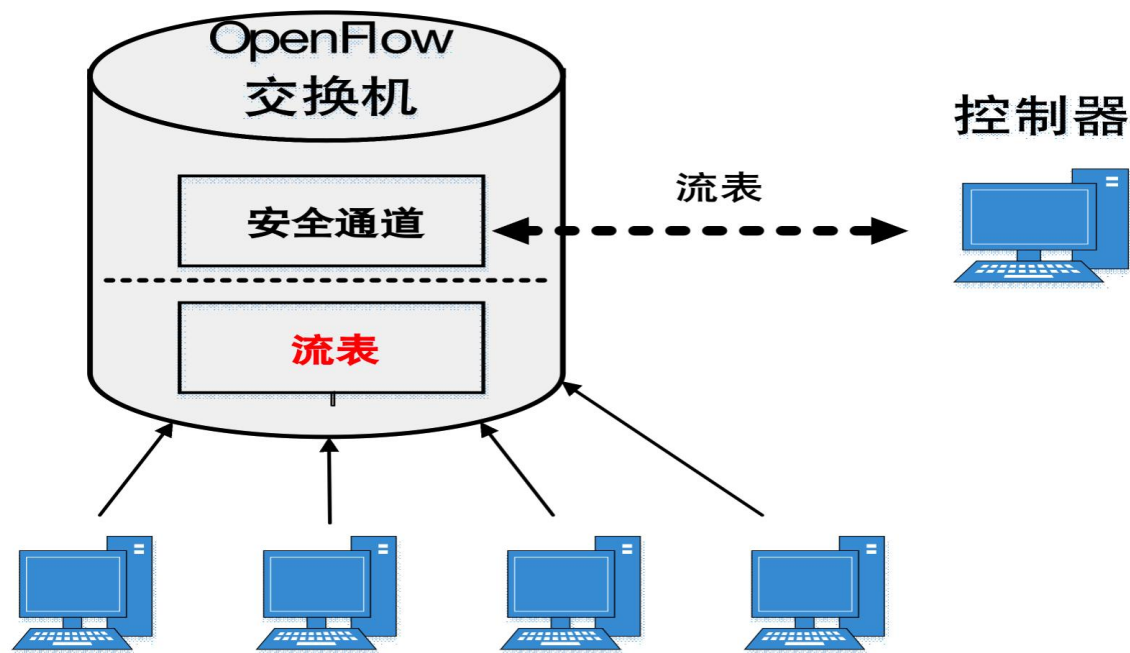
控制器:

控制器可以通过OpenFlow
协议就可以对网络交换设备
的流表 进行编程和管理。



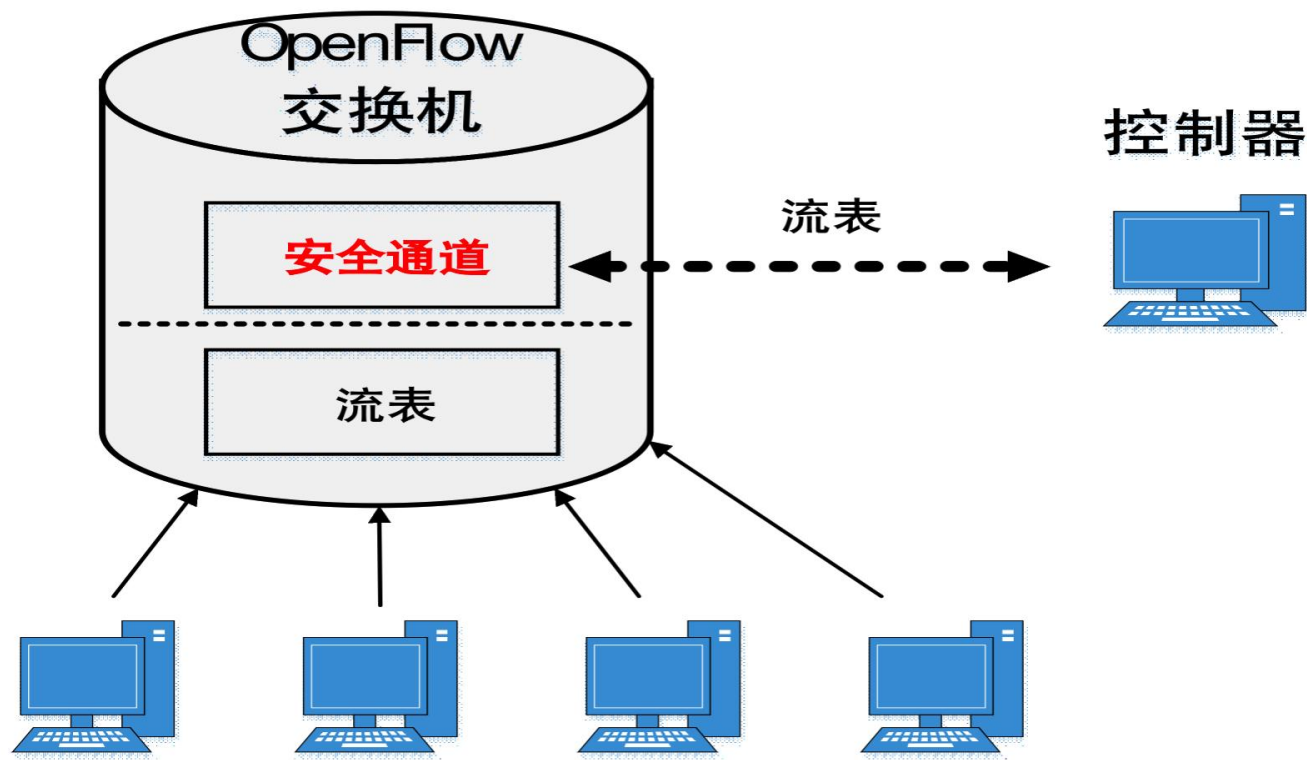
流表：

流表支持对远程的访问控制，
这样可以将流表的配置与管理
从网络交换设备本身中剥离出来，
也使得可以对整个网络中
流表 进行集中控制与管理，将
物理网络和逻辑网络有效的分
离开来



安全通道：

用于连接网络交换设备和远程网络控制器，在控制器和网络交换设备之间互相发送命令和数据包



场景回顾：

针对本小节开始提到的跨地域数据中心网络管理的案例，基于网络虚拟化的SDN技术可以允许网络工程师更为灵活地控制和管理网络，以便最好地服务需求，从而达到提高响应速度的效果。



02

桌面虚拟化

• 场景：云办公



办公系统都是部署在银行内网



在外出差的领导和外勤人员无法正常使用,工作效率不是太高

挑战：

客户急需一种能快速部署、平滑平移到平板电脑和手机上的移动办公产品,如何让**银行外的设备可以像在银行内部一样**,对内部系统的软件进行直接操作?

- 定义:

桌面虚拟化将用户的桌面环境和与其他的终端设备解耦合。服务器上面存放的是每个用户的完整桌面，用户可以通过任意的终端设备（如个人电脑、智能手机、PAD等），在任意时间，任意地点通过网络访问该桌面环境。

- 和云端的关系:

将来的终端各种各样，但只要前端采用了桌面虚拟化技术，用户能够在任何时间、任何地点、以各种方式对后端的云进行信息的处理与管理。



各种各样的终端产品。

- 场景回顾

针对本小节开始提到的远程云办公，该银行可采用面向桌面虚拟化技术的金山WPS云办公技术，利用移动终端进行云办公，原来需要在现场采集数据后回公司再次录入的工作现在已经可以在现场实时录入，大大减轻了工作人员的工作员，提高了工作效率。

常见的虚拟化产品

- Hyper-V虚拟化
- Xen虚拟化
- Vmware虚拟化
- VirtualBox虚拟化
- KVM虚拟化
- Docker虚拟化

06

容器技术

- 场景：平台虚拟化管理

某公司的平台上，一台 16 核 32G 内存的虚拟机上，需要跑 500+ 个用户的应用，有两个事情很重要：①资源隔离：比如限制应用最大内存使用量，或者资源加载隔离等；②低消耗：虚拟化本身带来的损耗需要尽量低。不可能在一台机器上开 500 个虚拟机，虽然可以在资源隔离方面做的很好，但这种虚拟化本身带来的资源消耗太严重。

如何在平台上，进行虚拟化的有效管理呢？

- 传统虚拟化技术的缺陷：

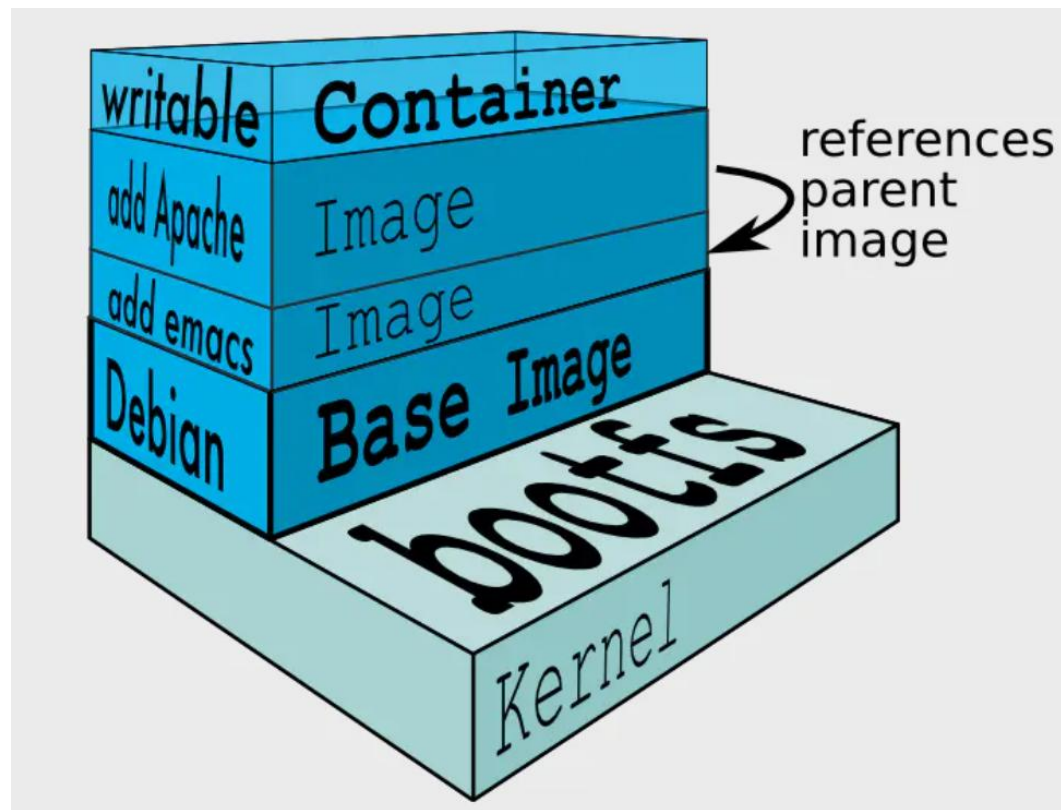
- 1.每一个虚拟机都是一个完整的操作系统，所以需要给其分配物理资源，当虚拟机数量增多时，操作系统本身消耗的资源势必增多
- 2.开发环境和线上环境的通常存在区别，所以开发环境与线上环境之间无法达到很好的桥接，在部署上线应用时，依旧需要花时间去处理环境不兼容的问题

- 新型虚拟化技术：容器

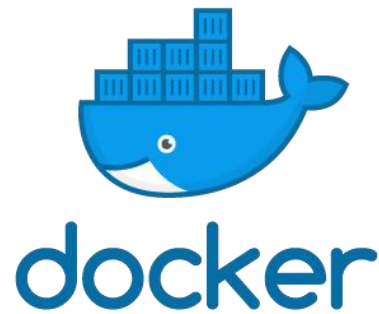
容器可以把开发环境及应用整个打包带走，打包好的容器可以在任何的环境下运行，这样就可以解决开发与线上环境不一致的问题了

- 代表应用：Docker

- 容器：利用一个开源的容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中，然后发布到任一Linux或Windows机器上，也可以实现虚拟化。
- 镜像：可执行的独立软件包，包含软件运行的内容：代码，运行时环境，系统工具，系统库和设置。

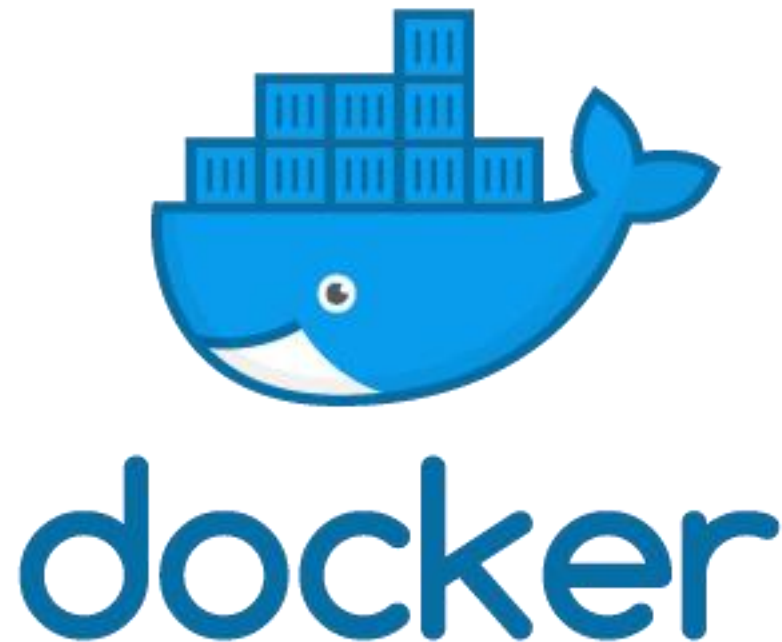


- Docker: 基于容器技术的轻量级虚拟化解决方案。Docker是容器引擎，把Linux的cgroup、namespace等容器底层技术进行封装抽象，为用户提供了创建和管理容器的便捷界面（包括命令行和API）
- 作用：将应用程序与该程序的依赖，打包在一个文件里，运行这个文件，就会生成一个虚拟容器。程序在这个虚拟容器里运行，就好像在真实的物理机上运行一样。有了Docker，就不用担心环境问题。
- 核心：实现应用与运行环境整体打包以及打包格式统一。

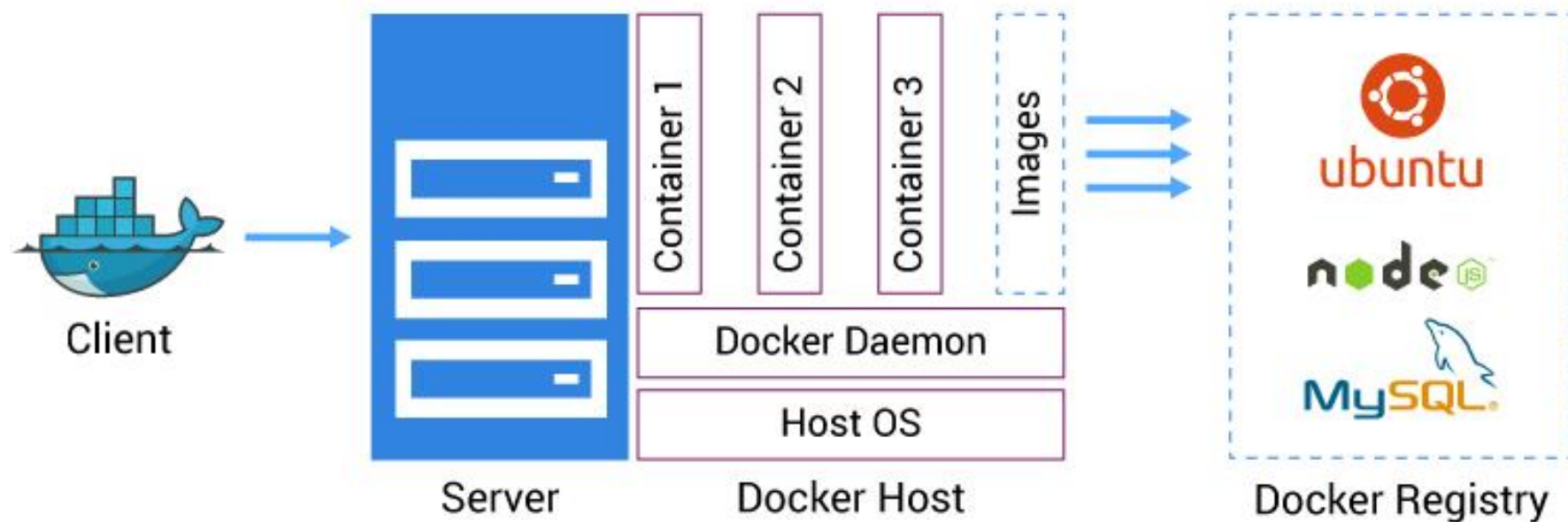


Docker带来的好处

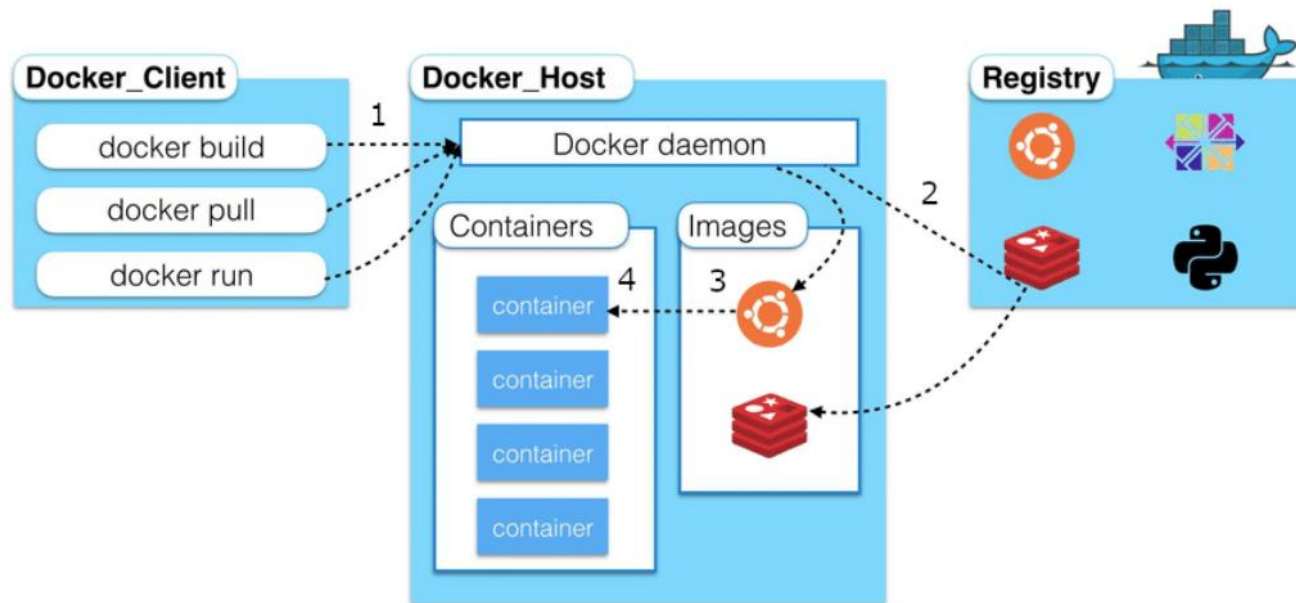
- 秒级的交付和部署
- 保证环境一致性
- 高效的资源利用
- 弹性的伸缩
- 动态调度迁移成本低



- 一个完整的Docker有以下几部分组成：
 - 客户端（Docker Client）、守护进程（Docker Daemon）、镜像（Docker Image）、容器（Docker Container）、仓库（Docker Registry）



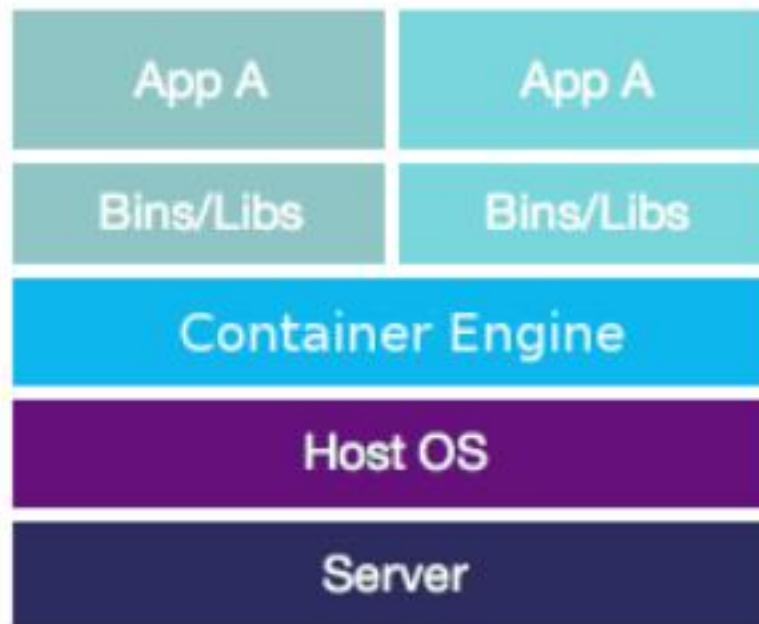
- Docker采用C/S架构，Docker Daemon作为服务端接受来自客户的请求，并处理这些请求（创建、运行、分发容器）。客户端和服务端既可以运行在一个机器上，也可通过socket或者RESTful API来进行通信。
- Docker daemon一般在宿主主机后台运行，等待来自客户端的消息。Docker客户端则为用户提供一系列可执行命令，用户用这些命令实现跟Docker daemon交互。



虚拟机与容器的对比



Virtual Machines



Containers

虚拟机与容器的对比

	容器技术	虚拟机技术
占用磁盘空间	小，甚至几十KB	非常大，上GB
启动速度	快，几秒钟	慢，几分钟
运行形态	直接运行在宿主机的内核上，不同容器共享同一个Linux内核	运行在Hypervisor上
并发性	一台宿主机可以启动成百上千个容器	最多几十个虚拟机
性能	接近宿主机本地进程	逊于宿主机
资源利用率	高	低

https://blog.csdn.net/newbie_807486852



Thank You!
感谢您的时间!

