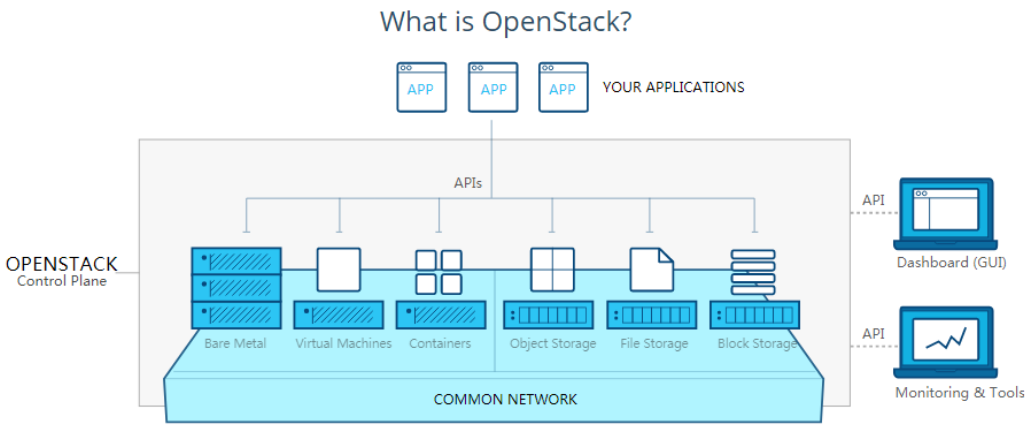
Openstack 介绍

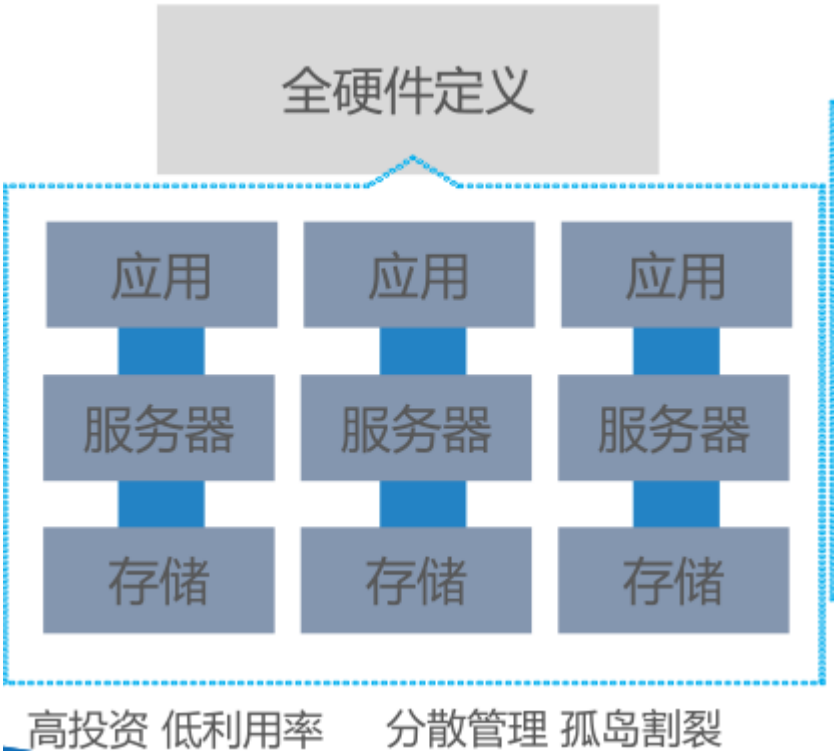


OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

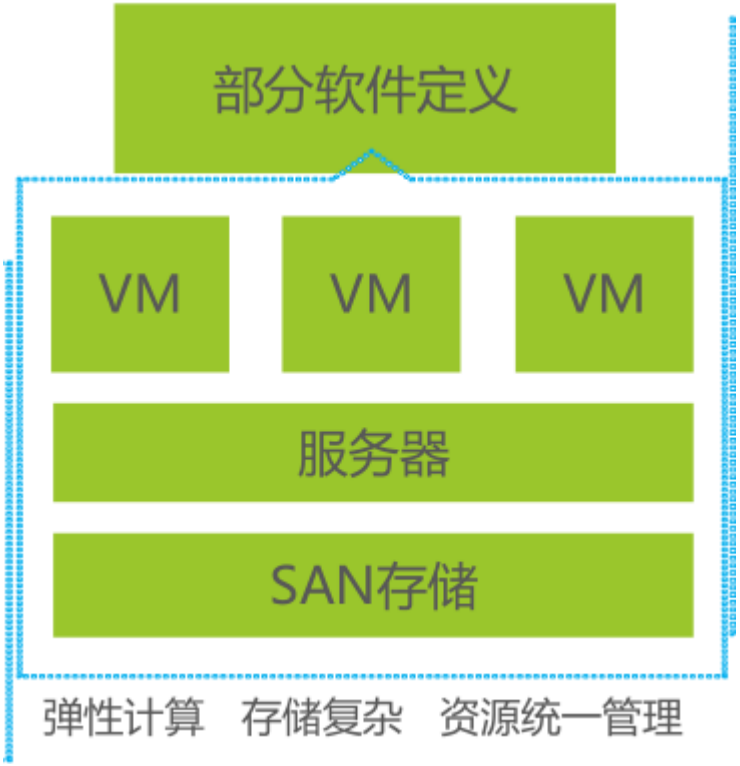
1 云计算基础

IAAS 层

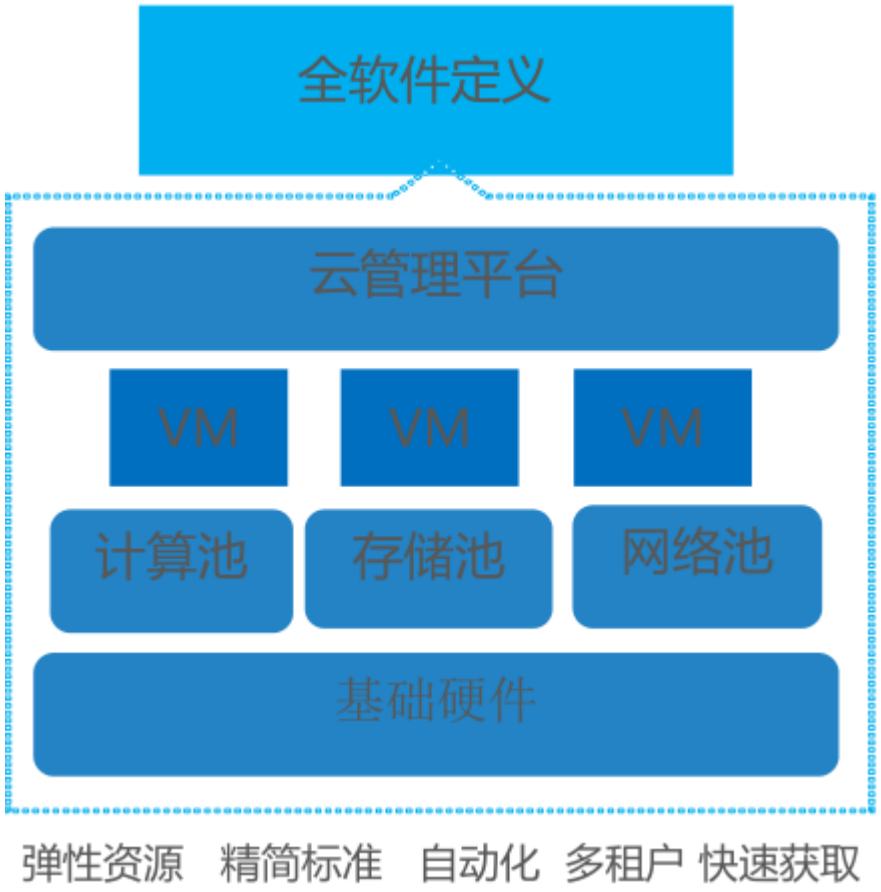
①全硬件定义



②部分软件定义



③全软件定义



2 虚拟化技术

2.0 基本介绍

虚拟化：计算资源的抽象和模拟

计算机资源

抽象和模拟：虚拟化的具体实现方法

- 平台虚拟机化，针对计算机和操作系统的虚拟化
- 资源虚拟化，针对特定的系统资源的虚拟机化，比如内存、存储、网络资源等
- 应用程序虚拟化，应用虚拟化是将应用程序与操作系统解耦合，为应用程序提供了一个虚拟机的运行环境。

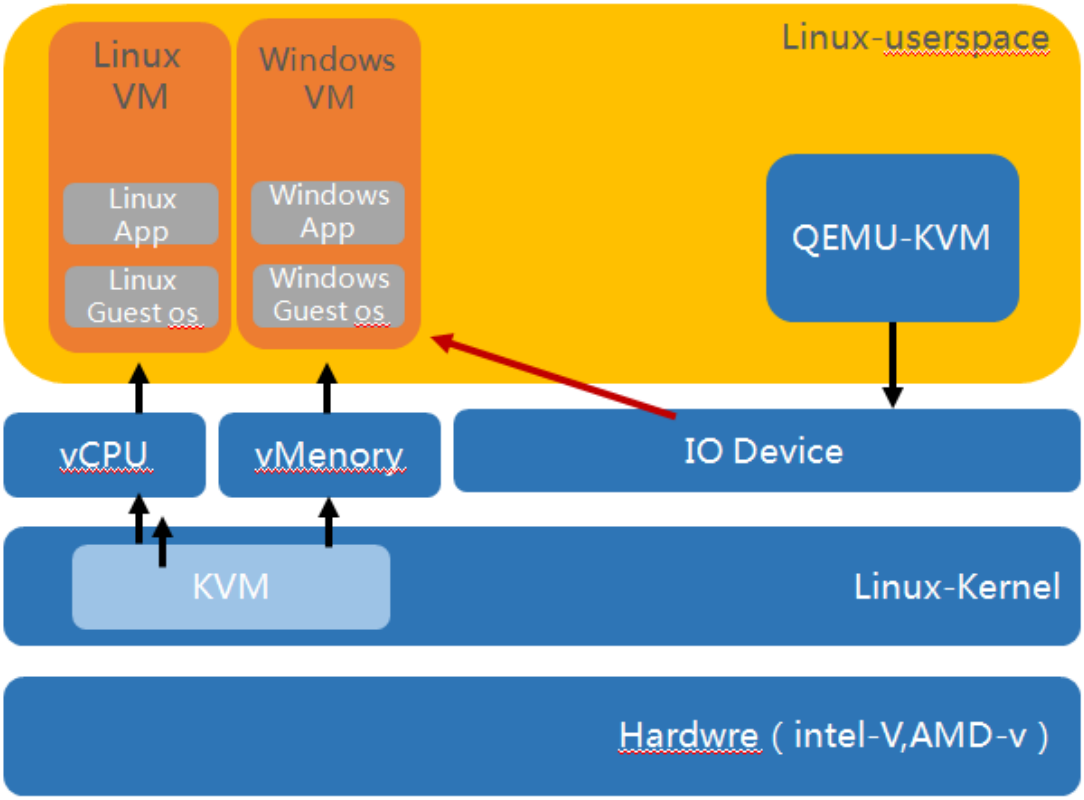
什么是 KVM

- KVM 全面是 Kernel-based Virtual Machine 的简称
- 基于硬件辅助的开源全虚拟化解决方案

KVM 的架构

- 基本上有两大方面组成，KVM 模块以及 QEMU-KVM
- Kvm.ko、kmv_intel.ko、kvm_amd.ko
- Qemu-kvm:通过修改 qemu 代码而得出的专门的管理工具和创建虚拟机的管理工具
- /dev/kvm:Linux 系统下 KVM 提供的驱动 接口

KVM 架构图

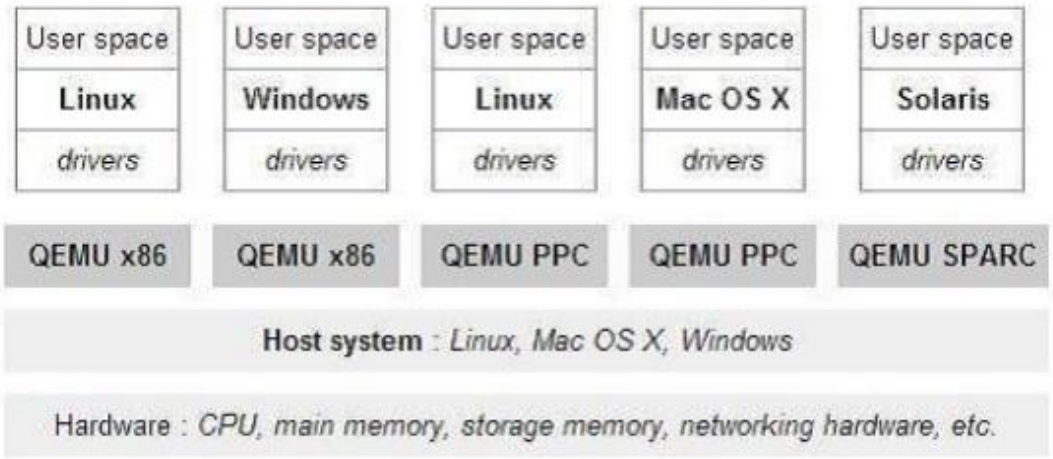


QEMU-KVM 与 KVM 关系流程

From https://www.cnblogs.com/echo1937/p/7138294.html

◆ What's QEMU

QEMU 是一个主机上的 VMM（virtual machine monitor）,通过动态二进制转换来模拟 CPU，并提供一系列的硬件模型，使 guest os 认为自己和硬件直接打交道，其实是同 QEMU 模拟出来的硬件打交道，QEMU 再将这些指令翻译给真正硬件进行操作。通过这种模式，guest os 可以和主机上的硬盘，网卡，CPU，CD-ROM，音频设备和 USB 设备进行交互。但由于所有指令都需要经过 QEMU 来翻译，因而性能会比较差：



◆ What's KVM?

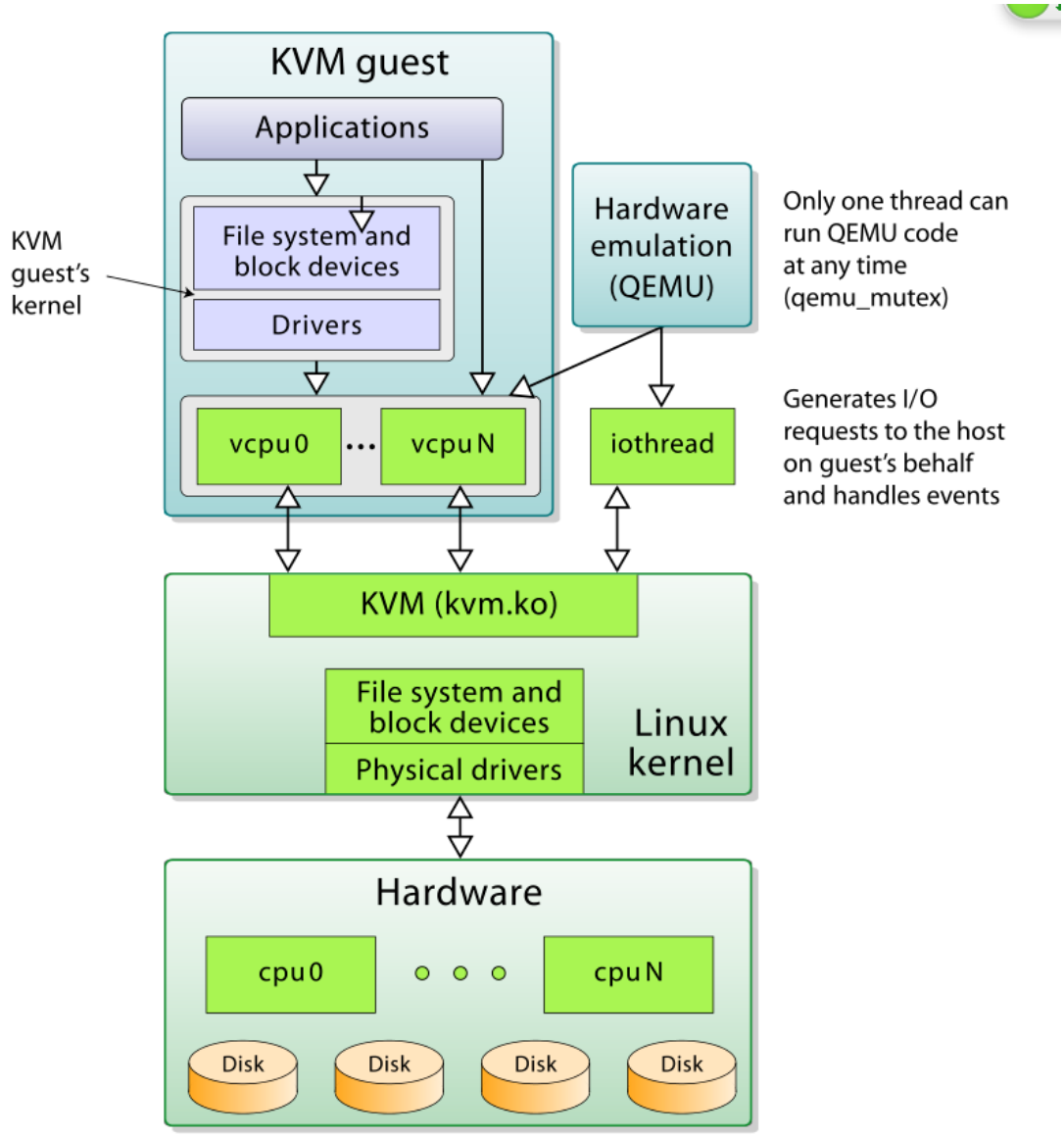
KVM 实际是 linux 内核提供的虚拟化架构，可将内核直接充当 hypervisor 来使用。KVM 需要处理器硬件本身支持虚拟化扩展，如 intel VT 和 AMD AMD-V 技术。KVM 自 2.6.20 版本后已合入主干并发行，除此之外，还以模块形式被移植到 FreeBSD 和 illumos 中。除了支持 x86 的处理器，同时也支持 S/390,PowerPC,IA-61 以及 ARM 等平台。

工作原理

KVM 包含一个内核模块 kvm.ko 用来实现核心虚拟化功能，以及一个和处理器强相关的模块如 kvm-intel.ko 或 kvm-amd.ko。KVM 本身不实现任何模拟，仅仅是暴露了一个/dev/kvm 接口，这个接口可被宿主机用来主要负责 vCPU 的创建，虚拟内存的地址空间分配，vCPU 寄存器的读写以及 vCPU 的运行。有了 KVM 以后，guest os 的 CPU 指令不用再经过 QEMU 来转译便可直接运行，大大提高了运行速度。但 KVM 的 kvm.ko 本身只提供了 CPU 和内存的虚拟化，所以它必须结合 QEMU 才能构成一个完整的虚拟化技术，也就是下面要介绍的技术。

◆ What's QEMU-KVM

从前面的介绍可知，KVM 负责 cpu 虚拟化+内存虚拟化，实现了 cpu 和内存的虚拟化，但 kvm 并不能模拟其他设备，还必须有个运行在用户空间的工具才行。KVM 的开发者选择了比较成熟的开源虚拟化软件 QEMU 来作为这个工具，QEMU 模拟 IO 设备（网卡，磁盘等），对其进行了修改，最后形成了 QEMU-KVM。



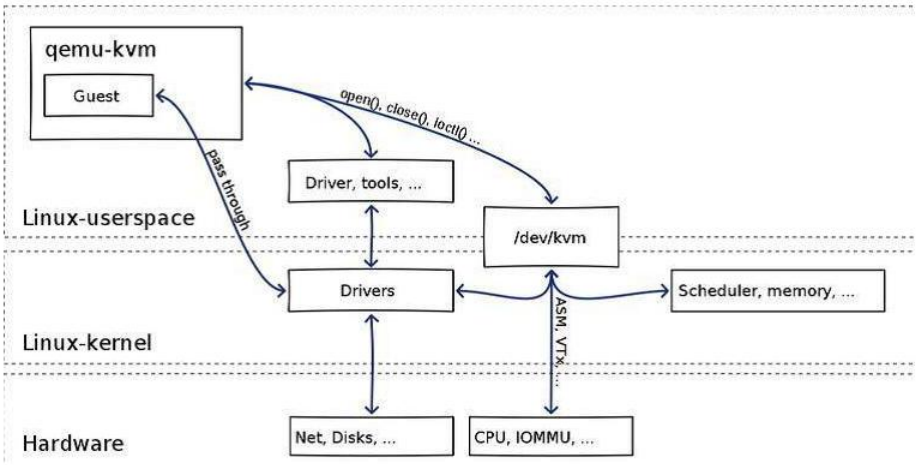
在 QEMU-KVM 中，**KVM 运行在内核空间，QEMU 运行在用户空间**，实际模拟创建、管理各种虚拟硬件，QEMU 将 KVM 整合了进来，通过/ioctl 调用 /dev/kvm，从而将 CPU 指令的部分交给内核模块来做，KVM 实现了 CPU 和内存的虚拟化，但 KVM 不能虚拟其他硬件设备，因此 qemu 还有模拟 IO 设备（磁盘，网卡，显卡等）的作用，KVM 加上 QEMU 后就是完整意义上的服务器虚拟化。

综上所述，QEMU-KVM 具有两大作用：

1 提供对 cpu，内存（KVM 负责），IO 设备（QEMU 负责）的虚拟

2 对各种虚拟设备的创建，调用进行管理（QEMU 负责）

这个方案中，QEMU 模拟其他的硬件，如 Network, Disk，同样会影响这些设备的性能。于是又产生了 pass through 半虚拟化设备 virtio_blk, virtio_net，提高设备性能。

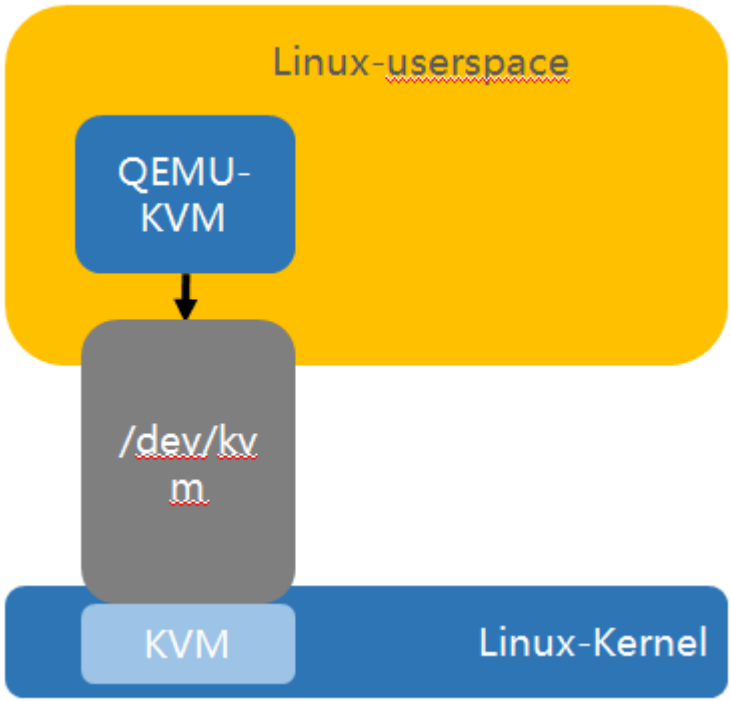


QEMU-KVM，是 QEMU 的一个特定于 KVM 加速模块的分支，里面包含了很多关于 KVM 的特定代码，与 KVM 模块一起配合使用。

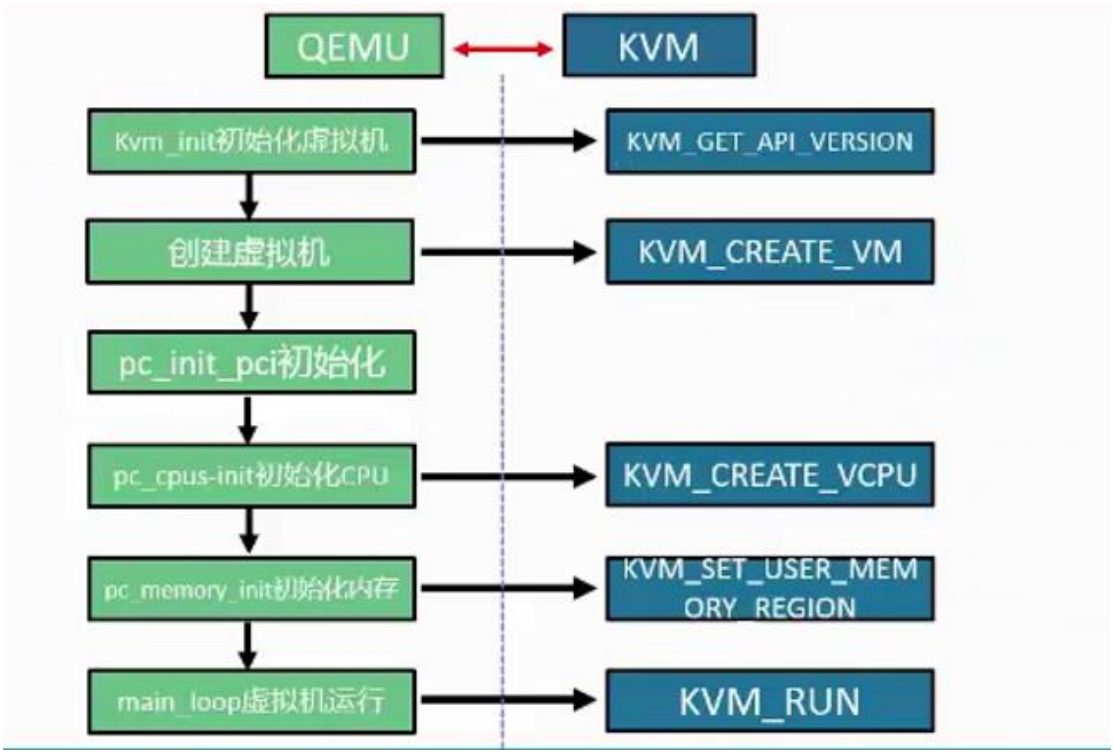
目前 QEMU-KVM 已经与 QEMU 合二为一，所有特定于 KVM 的代码也都合入了 QEMU，当需要与 KVM 模块配合使用的时候，只需要在 QEMU 命令行加上 --enable-kvm 就可以。

QEMU-KVM 与 KVM 关系流程

- Qemu-KVM 是 KVM 团队针对 qemu 改善和二次开发的一套工具
- /dev/kvm 是 kvm 内核模块提供给用户空间的一接口这个几口被 qemu-kvm 调用，通过 Ioctl 系统调用就可以用户提供一个工具用以创建，删除，管理虚拟机
- Qemu-kvm 就是通过 open（）close（）ioctl（）等方法去打开、关闭和调用这个接口实现跟 KVM 的互动

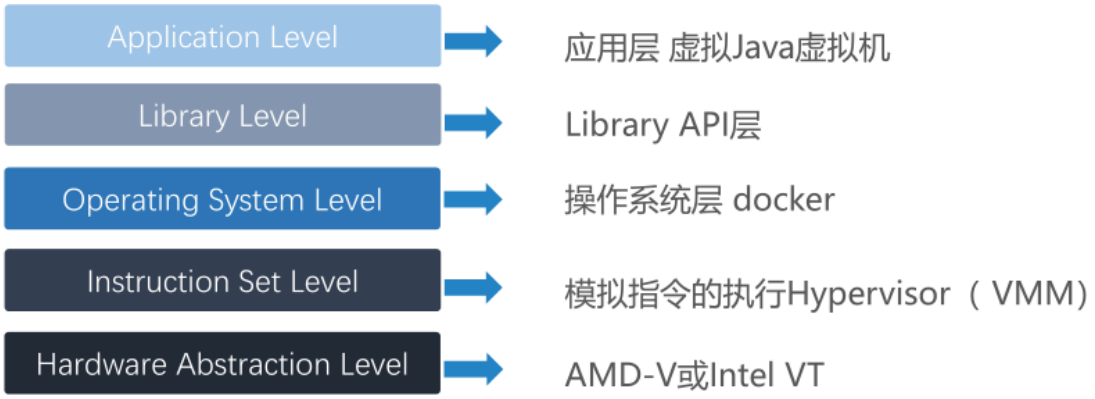


QEMU-KVM 与 KVM 关系流程



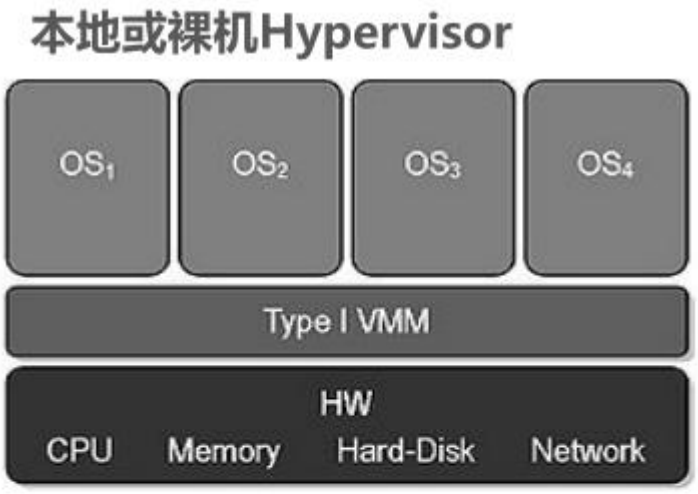
2.1 Kernel-based Virtual Machine(KVM)

虚拟化是一种资源管理技术，将计算机的各种实体资源(CPU,内存，磁盘空间，网络适配器等)，予以抽象、转换后呈现出来并可供分区、组合为一个或多个



2.2 虚拟化抽象分 5 个层次

层次	例子
Application Level 应用程序级	JVM/.Net CLR
Library Level 库支持（用户级 API）级	WINE ubuntu 虽然是一个独立的系统，不兼容任何其他格式的应用。但作为一个刚刚受人们关注的系统。人们能使用的应用还是较少。而 wine 能模拟打开 Windows 应用，即 EXE 文件。以此解决一些，Ubuntu 不能解决，但 window 可以解决的问题。 https://jingyan.baidu.com/album/c1a3101e8e6ec7de646deb68.html?picindex=6
Operation System Level 操作系统级	Docker
Instruction Set Level 硬件抽象级	VMware/Xen/KVM
Hardware Abstraction Level 指令集体系结构级	Bochs

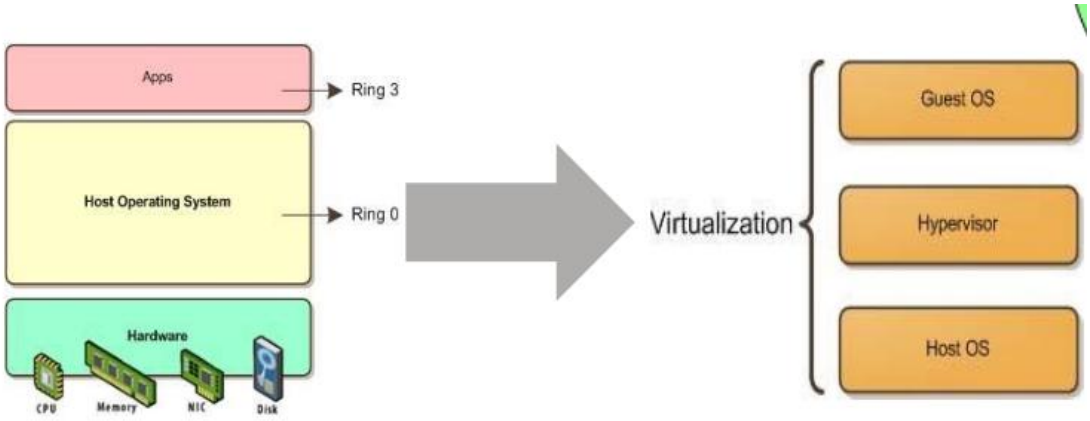


运行效率高、安全稳定可靠

1.VMware5.5及以后版本
2.Xen3.0以后版本
3.Virtual PC 2005
4.KVM

2.3 安全分级保护

分级保护域：一种用来在发生故障时保护数据和功能，提升容错度，避免恶意操作，提升计算机安全的设计方式。



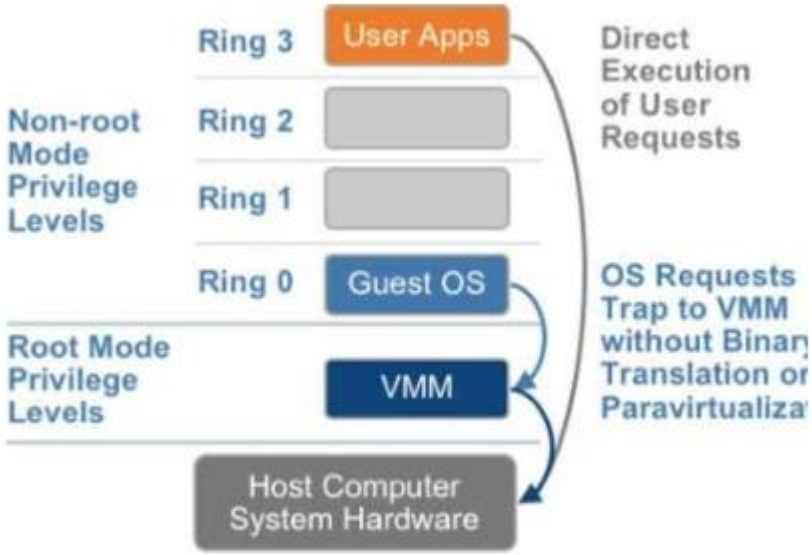
2.4 内核态 KVM

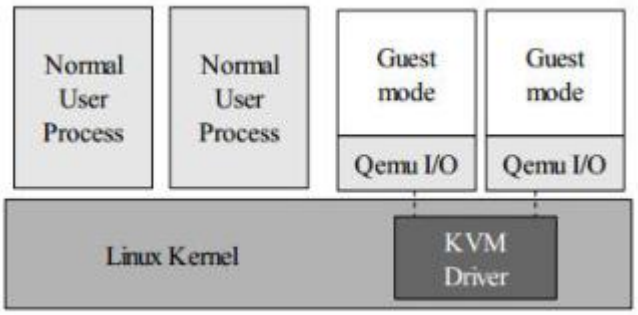
KVM 是 linux 内核中的虚拟化基础设施，可以将 Linux 内核转化为一个 Hypervisor

KVM 是内核的模块，采用硬件辅助虚拟化技术 Intel-VT,AMD-V

Guest OS 的 CPU 指令不用经过转译，直接运行，大大提高了速度

进程独立，模式分工，相关隔离，整体安全性非常高





2.5 目前主流的虚拟化公司

名称	归属	主CPU	目标CPU	主系统
Hyper-V	微软	x86-64+硬件辅助虚拟（AMD-V或Intel VT）	x86-64，x86	Windows Server 2008, Windows Server 2012, Windows 8
KVM	Red Hat	Intel/AMD处理器、x86虚拟化	x86-64，x86	Linux（内核级）
VMware ESXi Server	VMware	Intel x86，x86-64	x86，x86-64	裸机安装（内核级）
VMware Workstation	VMware	Intel/AMD x86，x86-64	x86，x86-64	Windows，Linux
Xen	citrix	Intel x86，x86-64	x86，x86-64	NetBSD，Linux，Solaris

2.6 关键术语

VMM (Virtual Machine Monitor)；Hypervisor	快照/克隆（Snapshot/clone）
Guest OS	热迁移；冷迁移
Host OS; HostOS 磁盘	操作系统镜像
HA （High Availability）	云硬盘（storage），云主机（VM）
DRS/DPM Distributed Resource Scheduler/ Distributed Power Management 分布式资源调度/分布式能源管理	多租户（multi-tenancy ）
P2V/V2V	
DVS(distributed virtual switch)	
VDI(Virtual Desktop Infrastructure)/VOI(Virtual OS Infrastructure)	

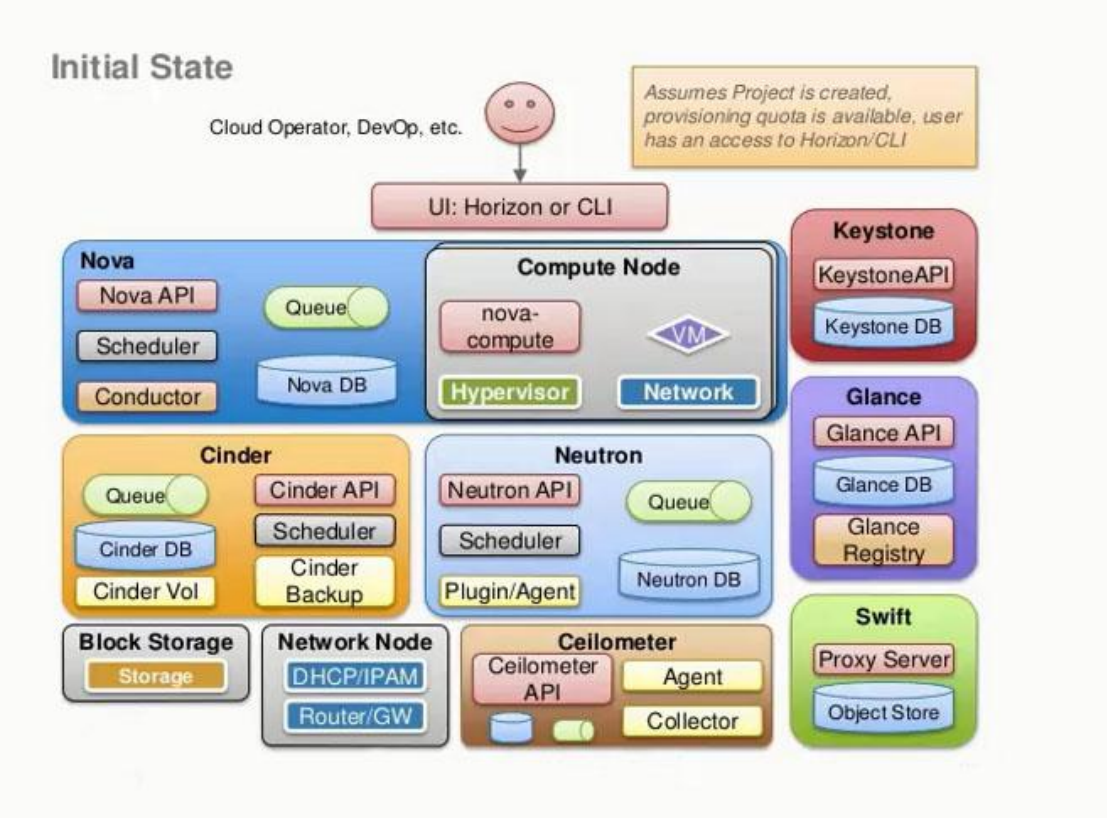
2.7 平台技术模块：Openstack 技术

2.7.1 openstack 架构及核心服务

核心服务

Nova，keystone，neutron，cinder 等

- Keystone：负责认证
- Nova：负责计算
- Glance：负责镜像
- Neutron：负责网络
- Swift：负责对象存储
- Ceilomenter：负责监控
- openstack cinder :块存储服务



2.7.2 Openstack----Nova

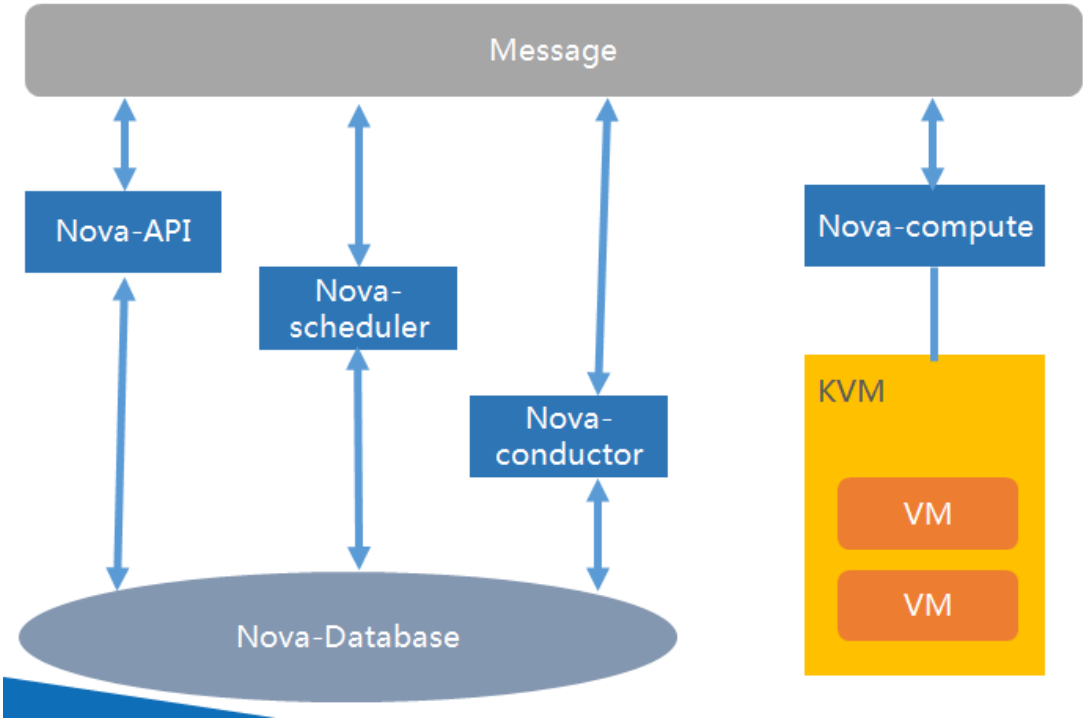
From https://docs.openstack.org/nova/pike/

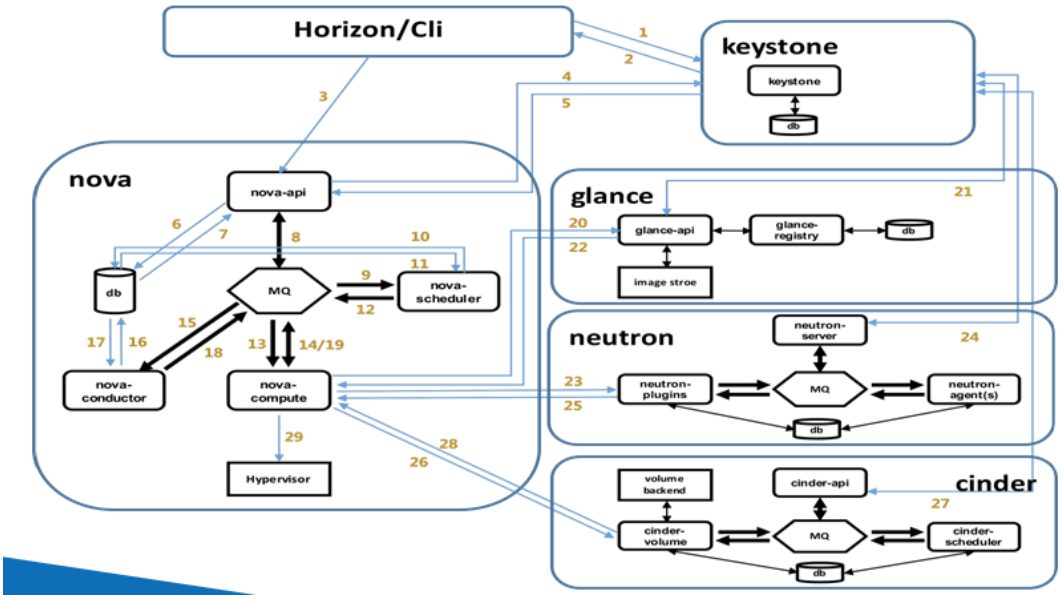
Nova is the OpenStack project that provides a way to provision compute instances (aka virtual servers). Nova supports creating virtual machines, baremetal servers (through the use of ironic), and has limited support for system containers.

Nova runs as a set of daemons on top of existing Linux servers to provide that service.

It requires the following additional OpenStack services for basic function:
(1)Keystone: This provides identity and authentication for all OpenStack services.
(2)Glance: This provides the compute image repository. All compute instances launch from glance images.
(3)Neutron: This is responsible for provisioning the virtual or physical networks that compute instances connect to on boot.
It can also integrate with other services to include: persistent block storage, encrypted disks, and baremetal compute instances.

- 管理 instance 生命周期
- 生成、调度、终止实例





2.7.3 Openstack----Keystone

Keystone is an OpenStack service that provides API client authentication, service discovery, and distributed multi-tenant authorization by implementing OpenStack's Identity API.

- 管理用户及其权限
- 维护 Openstack 各项服务
- Authentication（认证）和 Authorization（授权）
- User：使用 openstack 的实体，用户，系统或者服务
- Credentials：身份证明信息，用户名密码，token，API key 等
- Authentication：Keystone 验证 user 身份的过程
- Token：数字和字母组成的字符串，默认有效期是 24 小时
- Project：对 openstack 资源进行分组和隔离
- Service:每个组件都差不多是一个 service，比如 Nova，Cinder，Neutron

User	住宾馆的人
Credentials	身份证，VIP贵宾卡
Authentication	入住登记过程，比如出示你身份证和VIP贵宾卡给前台
Token	宾馆给你的门卡（有时效性，住一天的话就是24小时）
Project	宾馆能提供的服务套餐项目
Service	宾馆可以提供的服务类别，比如，饮食类，娱乐类

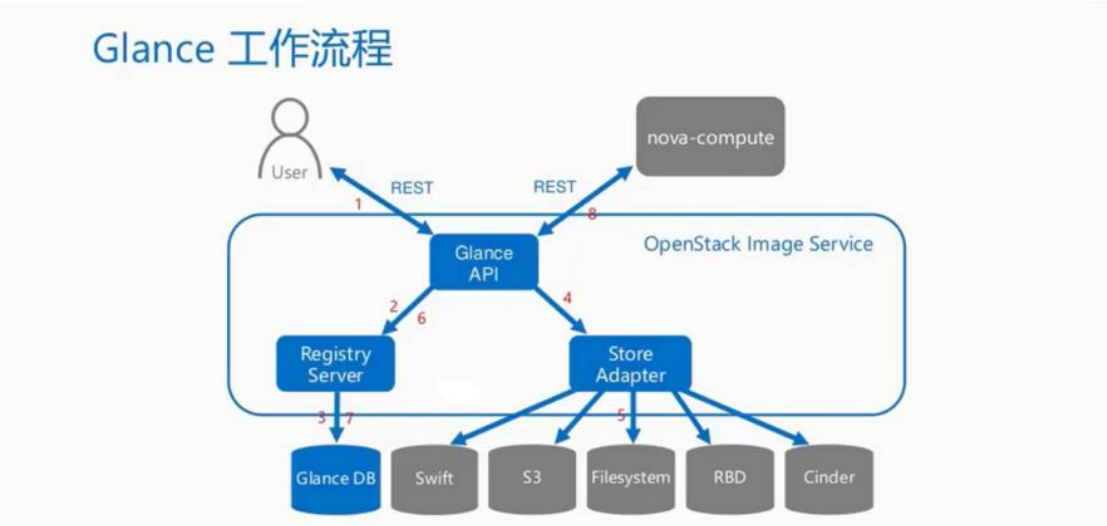
2.7.4 Openstack----Glance

From https://docs.openstack.org/glance/pike/

The Image service (glance) project provides a service where users can upload and discover data assets that are meant to be used with other services. This currently includes images and metadata definitions.

Glance image services include discovering, registering, and retrieving virtual machine (VM) images. Glance has a RESTful API that allows querying of VM image metadata as well as retrieval of the actual image.

- 提供镜像模板
- 支持 Ceph，Swift 存储等



2.7.5 Openstack----Neutron

From https://wiki.openstack.org/wiki/Neutron

Neutron is an OpenStack project to provide "network connectivity as a service" between interface devices (e.g., vNICs) managed by other OpenStack services (e.g., nova). It implements the Neutron API.

From https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/common/get_started_networking.html

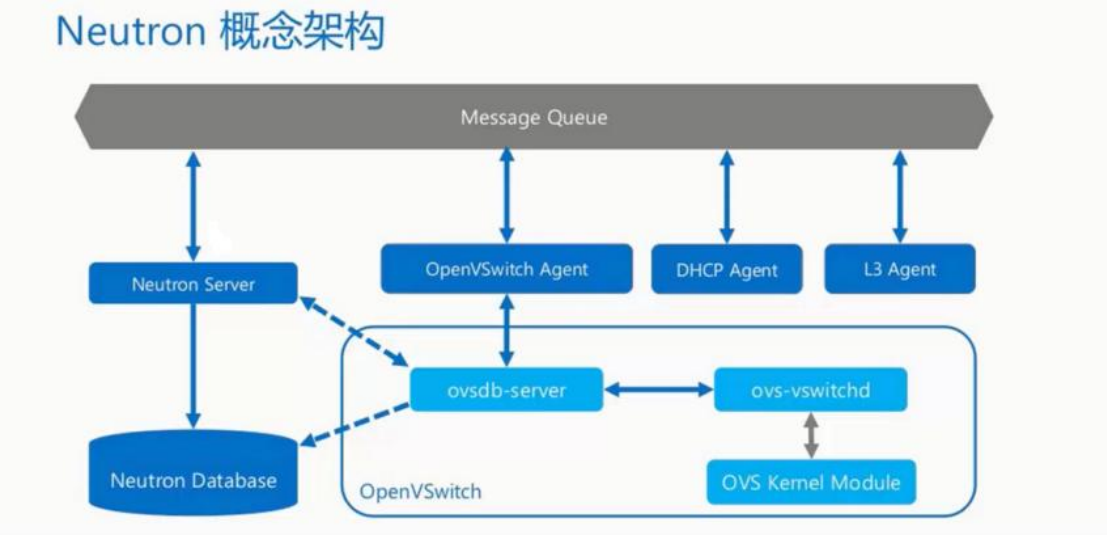
From https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/neutron-concepts.html

OpenStack Networking（neutron），允许创建、插入接口设备，这些设备由其他的 OpenStack 服务管理。插件式的实现可以容纳不同的网络设备和软件，为 OpenStack 架构与部署提供了灵活性。

- ◆ 提供网络服务
- ◆ 可插拔的网络架构设计
- ◆ 支持众多主流网络供应商
- ◆ SDN

Neutron 提供的网络服务？

- ◆ 二层交换
- ◆ 三层路由功能
- ◆ 负载均衡
- ◆ 防火墙


Neutron 概念架构
Message Queue
Neutron Server
OpenVSwitch Agent
DHCP Agent
L3 Agent
ovsdb-server
ovs-vswitchd
Neutron Database
OpenVSwitch
OVS Kernel Module

3 企业级存储

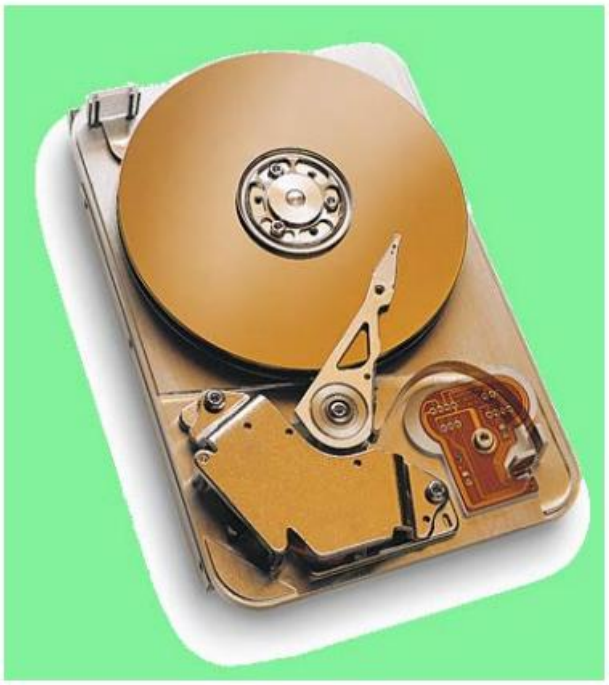

串行接口
空气过滤片
磁盘
主轴（下方是
轴承和马达电机）
磁头
磁头停泊区
音圈马达
磁头臂
永磁铁

3.1 硬盘类型介绍

硬盘类型	转速	MTBF	IOPS	持续读写速率	应用特点
SATA/NL-SAS	7200 RPM	120万小时	80-100	120-160MB/s	大容量，低转速，性价比高
SAS硬盘	10000 RPM	160万小时	120-150	120-200MB/s	可靠，高性能
SAS硬盘	15000 RPM	160万小时	180-200	120-200MB/s	可靠，高性能
SSD硬盘	无	200万小时	5万-10万	100-500MB/s	小容量，超高性能

3.2 硬盘指标-关键参数介绍

- 接口速率
- 容量
- 尺寸
- 主轴转速



3.3 RAID

RAID（Redundant Array of Independent Disks）：独立冗余磁盘组。其基本思想就是把多个相对便宜的硬盘组合起来，成为一个硬盘阵列组，使性能达到甚至超过一个价格昂贵、容量巨大的硬盘。

3.4 常用 RAID 级别比较

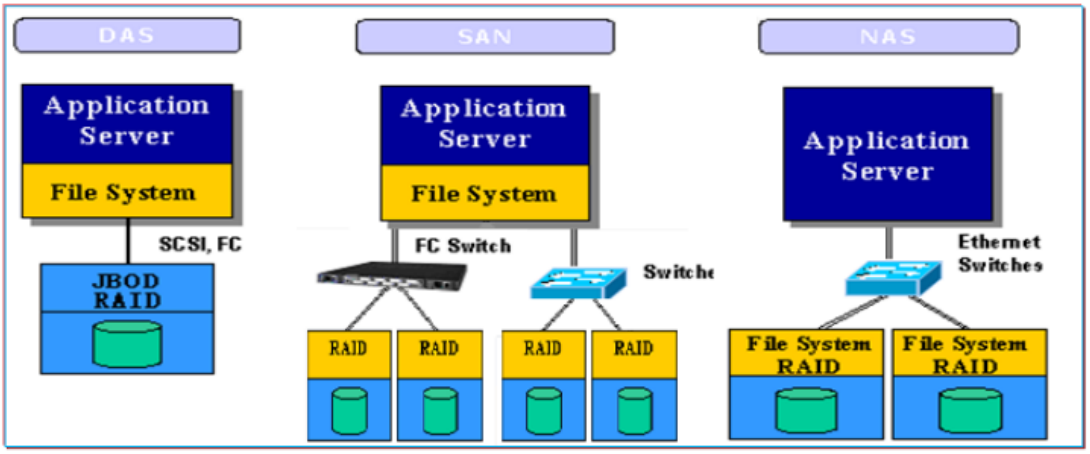
RAID级别	RAID 0	RAID 1	RAID 3	RAID 5	RAID 1+0
别名	条带	镜像	专用奇偶位条带	分布奇偶位条带	镜像数组条带
容错性	无	有	有	有	有
冗余类型	无	复制	同位	同位	复制
热备盘选项	无	有	有	有	有
读性能	高	低	高	高	一般
随机写性能	高	低	最低	低	一般
连续写性能	高	低	低	低	一般
最小硬盘数	2块	2块	3块	3块	4块
可用容量	N * 单块	(N /2) * 单块硬盘容量	(N -1) * 单块硬盘容量	(N -1) * 单块硬盘容量	(N /2) * 单块硬盘容量

3.5 存储架构分类

DAS (Direct Attached Storage)直接挂接存储

SAN (Storage Area Network)存储区域网络 (包括 FC SAN、IP SAN)

NAS (Network Attached Storage)网络挂接存储



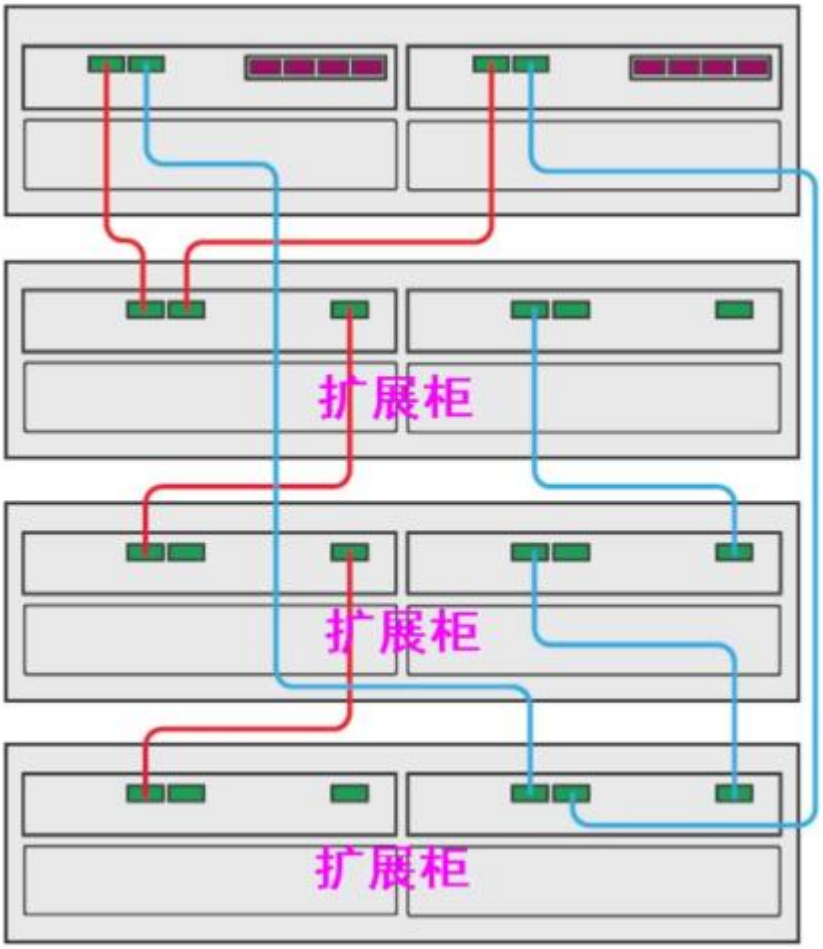
NAS 以文件的形式+LAN 连接存储介质；

而 SAN 以块形式+光纤连接存储介质。

3.6 传统存储架构的不足

传统架构存储是对称多处理器,通常是双控存储

缺点:控制器扩展能力有限,导致性能、容量有扩展瓶颈



性能瓶颈

扩展容量不能与性能线性增长，扩展容量越大性能越差

管理复杂

异构存储设备，每个设备都需要专业的维护人员

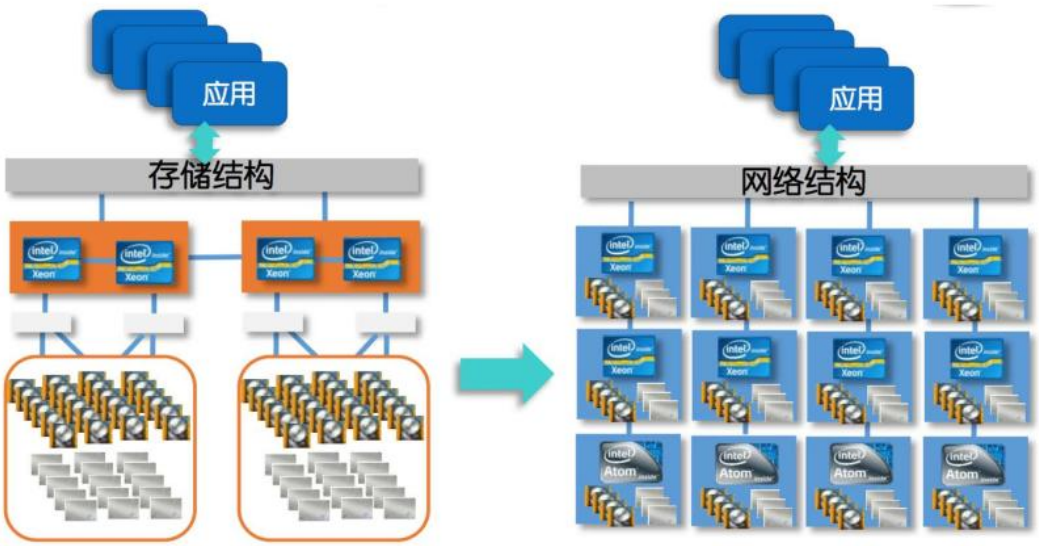
难以扩展

SAN 存储采用 Scale-up 扩展，无法预知能够扩展的规模

价格昂贵

SAN 存储采用专用设备，采购和维护成本昂贵。

3.7 分布式存储



Ceph 是一个高可用、易于管理、开源的分布式存储系统，可以在一套系统中同时提供**对象存储、块存储以及文件存储服务**。其主要由 Ceph 存储系统的核心 RADOS 以及块存取接口、对象存取接口和文件系统接口组成

3.8 存储技术模块：Ceph 存储技术

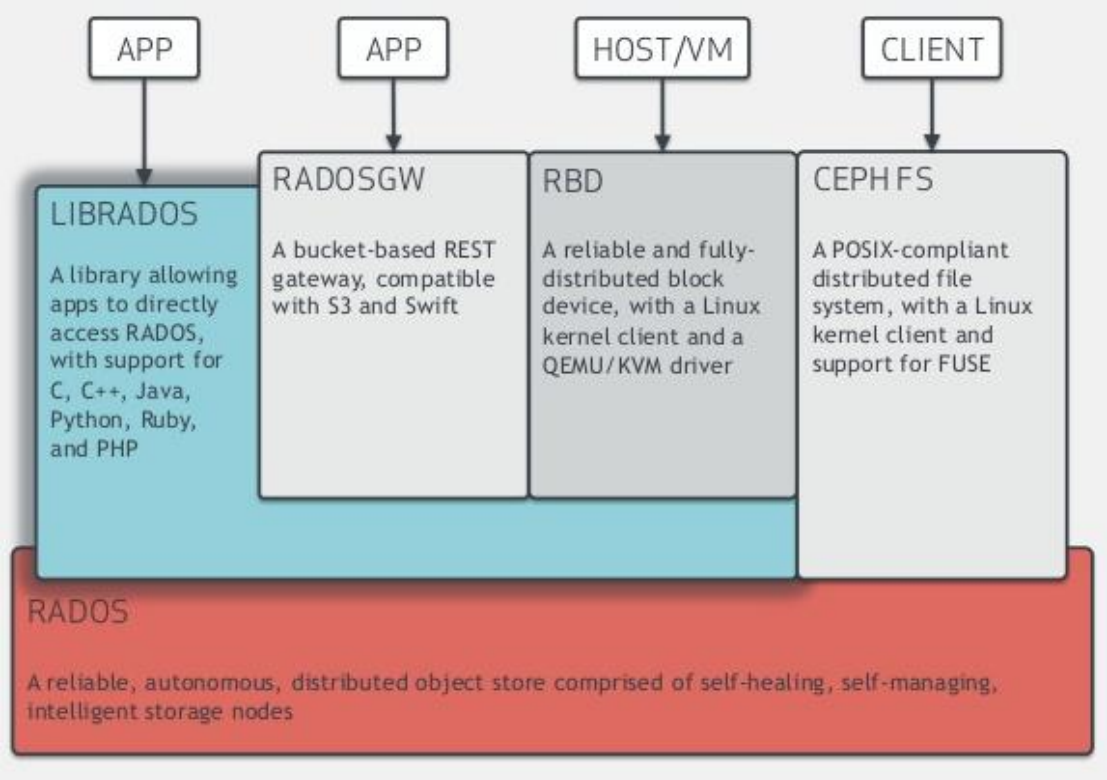
3.8.1 ceph 的架构

From http://docs.ceph.org.cn/

From https://ceph.com/

< Ceph 分布式存储学习指南>

ceph 是一个开源的 PB 级文件系统，最早是加州大学 Santa Cruz 分校的一个研究项目,项目创始人 sage weil 是该校的一名博士。 ceph 包括一个兼容 POSIX 的分布式文件 CephFS，一个分布式对象存储系统 RADOS(ReliableAutonomic Distributed Object Storage),并基于 RADOS 实现了一个且兼容 Swift 和 S3 的存储系统 radosgw，以及一个块设备驱动 RBD

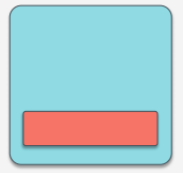


3.8.2 使用 ceph rbd

- mon 和 osd

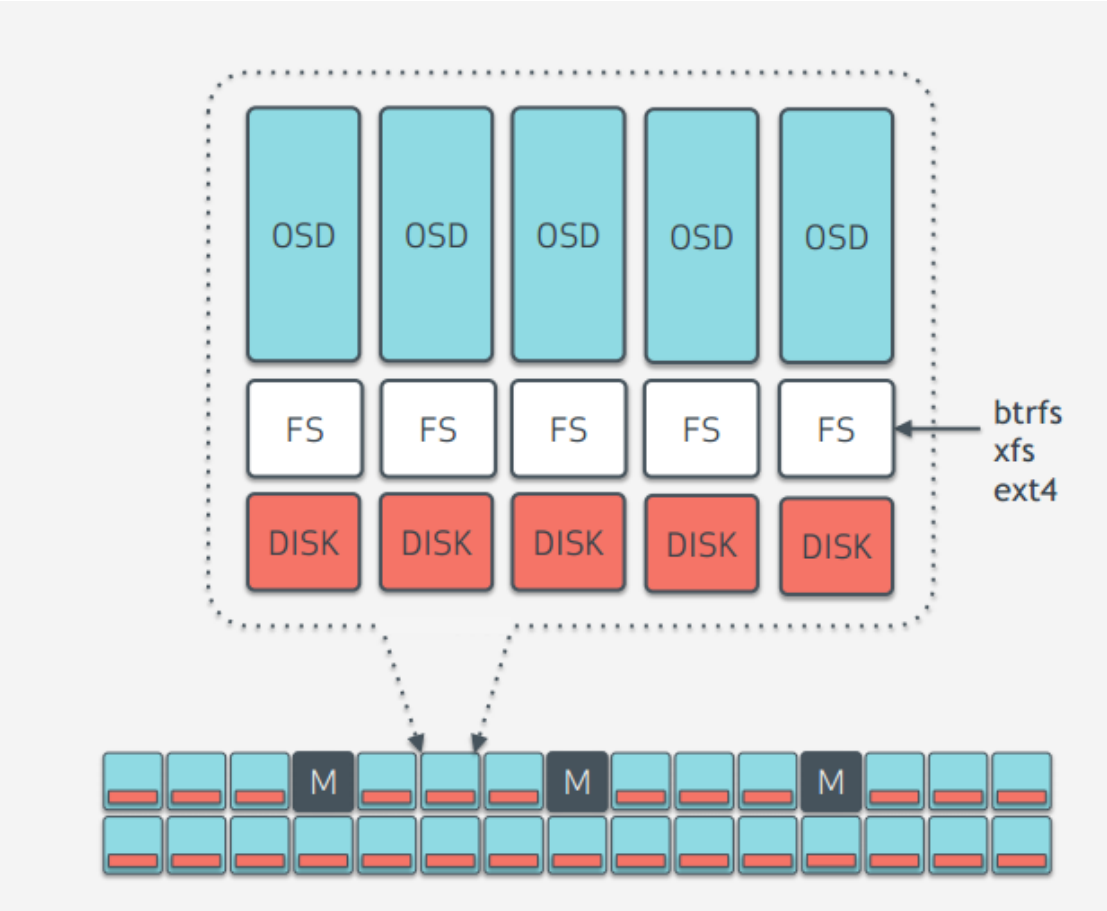
mon：

维护着展示集群状态的各种图表，包括监视器图、 OSD 图、归置组（ PG ）图、和 CRUSH 图。 Ceph 保存着发生在 Monitors 、 OSD 和 PG 上的每一次状态变更的历史信息



osd：

存储数据，处理数据的复制、恢复、回填、再均衡，并通过检查其他 OSD 守护进程的心跳来向 Ceph Monitors 提供一些监控信息



From https://www.cnblogs.com/yue-hong/p/7170263.html

◆ pg

crush:

通过伪随机算法来确保均匀的数据分布

Ojbect 通过一致性 HASH 算法，存入 PG 中

– oid=ino+ono

– hash（oid）& mask -> pgid

PG 通过 CRUSH 算法，确定要放在哪个 OSD 中

– CRUSH(pgid) -> osd1,osd3


对象
对象: 10
对象: 00
对象: 11
hash(object name) % num PG
PG
Pool 1
10
00
11
CRUSH(pg, cluster state, policy)
OSD
10
10 00
11 00
00
11
对象:00
Client

3.8.3 读写流程

◆ 写流程

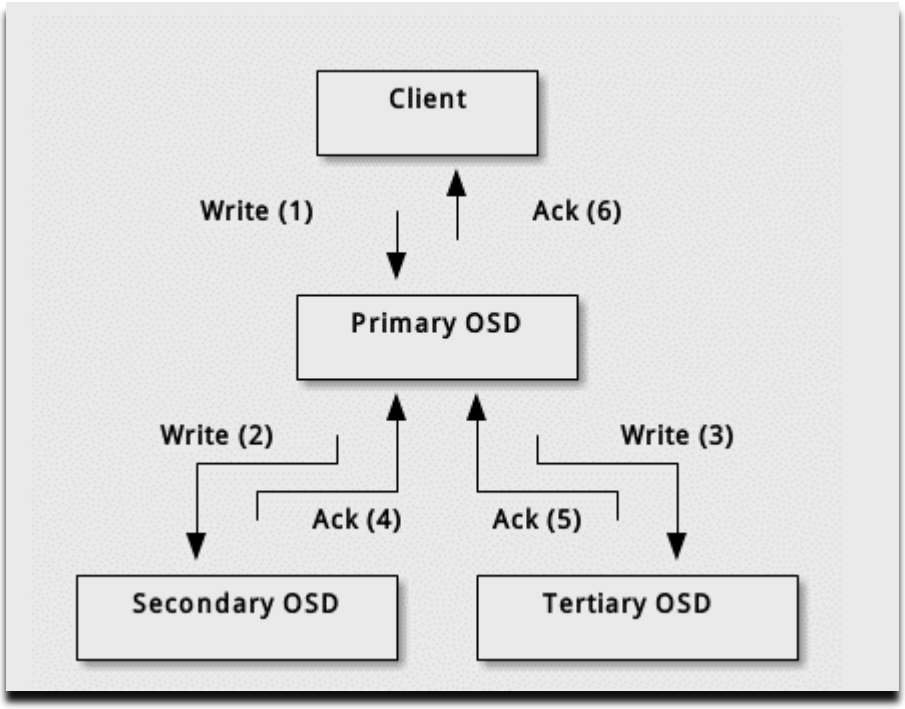
1.客户端会将数据发送给主 osd

2.主 osd 要先进行写操作预处理，完成后它要发送写消息给其他的从 osd，让他们对副本 pg 进行更改

3.主从 osd 通过 FileJournal 完成写操作到 Journal 中。从 osd 发送消息告诉主 osd 完成 journal，从 osd 进入 5

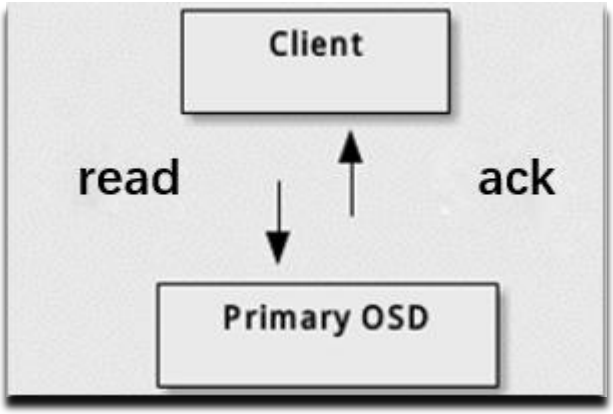
4.主 osd 收到所有的从 osd 完成写操作的消息后，会通知客户端，已经完成了写操作。主 osd 进入 5

5.主 osd，从 osd 的线程开始工作调用 Filestore 将 Journal 中的数据写入到底层文件系统中。主 osd 收到所有的从 osd 完成写 Journal 操作的消息后，会通知客户端数据可读。



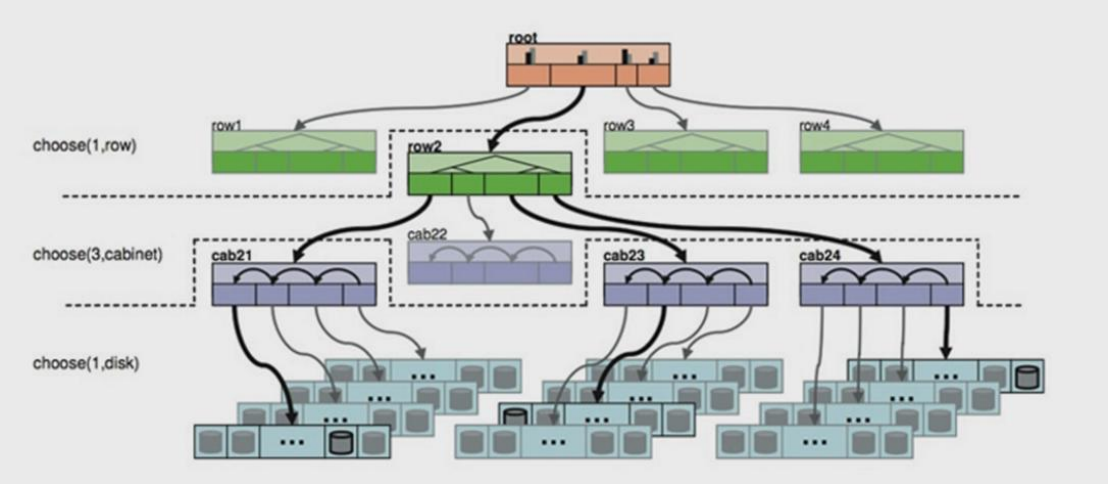
◆ 读流程

读取数据，client 只需完成同样的寻址过程，并直接和 Primary OSD 联系。目前的 Ceph 设计中，被读取的数据仅由 Primary OSD 提供



3.8.4 crush map 浅析

CRUSH 图包含 OSD 列表、把设备汇聚为物理位置的“桶”列表、和指示 CRUSH 如何复制存储池里的数据的规则列表。由于对所安装底层物理组织的表达， CRUSH 能模型化、并因此定位到潜在的相关失败设备源头，典型的源头有物理距离、共享电源、和共享网络，把这些信息编码到集群运行图里， CRUSH 归置策略可把对象副本分离到不同的失败域，却仍能保持期望的分布。例如，要定位同时失败的可能性，可能希望保证数据复制到的设备位于不同机架、不同托盘、不同电源、不同控制器、甚至不同物理位置



实例

weight

Ceph 用双整形表示桶权重。权重和设备容量不同，我们建议用 1.00 作为 1TB 存储设备的相对权重，这样 0.5 的权重大概代表 500GB 、 3.00 大概代表 3TB 。较高级桶的权重是所有枝叶桶的权重之和。

CRUSH 算法根据种每个设备的权重尽可能概率平均地分配数据

```
# buckets
host sata01 {                指定其类型、惟一名称（字符串）
    id -2                    惟一负整数 ID
    # weight 18.100          指定和各条目总容量/能力相关的权重
    alg straw                指定桶算法（通常是 straw ）
    hash 0  # rjenkins1      哈希（通常为 0 ，表示哈希算法 rjenkins1
    item osd.1 weight 1.810 指定枝叶桶，指定定和各条目总容量/能力相关的权重

}
```

3.8.5 ceph striping

3.8.6 rbd cache

4 虚拟化网络(需要反思)

https://www.cnblogs.com/pmyewei/p/6280445.html?utm_source=itdadao&utm_medium=referral

4.1 网络 OSI 七层模型

4.2 数据中心网络架构

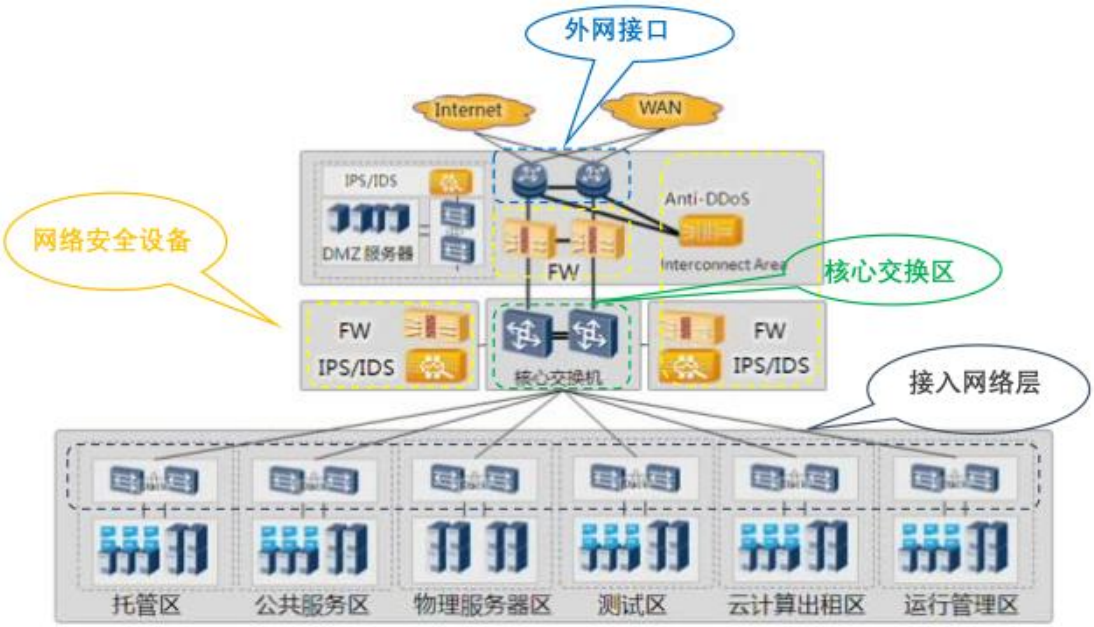
数据中心网络包含：

接入网络层，如：二层交换机或三层交换机。

核心交换区（核心交换机）

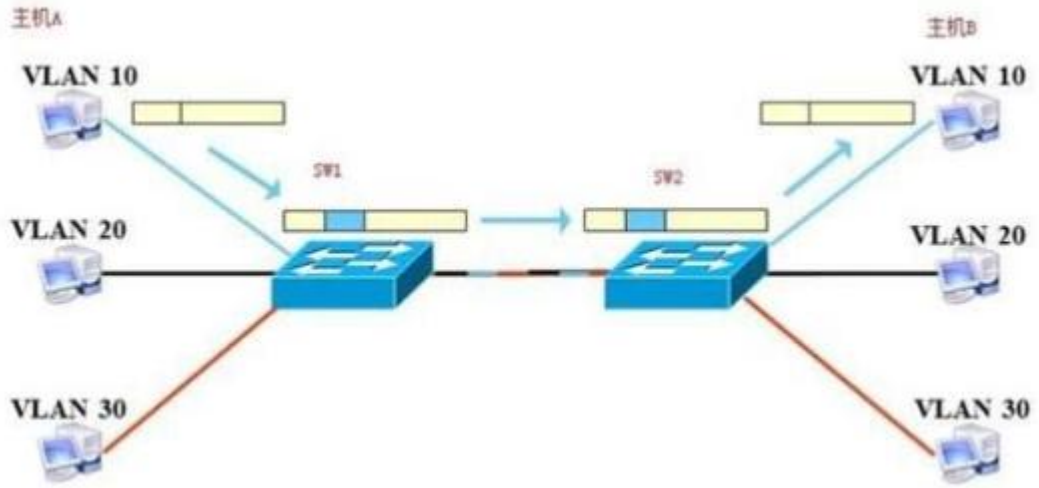
网络安全设备，如防火墙、IPS、IDS

外网接口，如路由器等组成



4.3 局域网下的业务隔离

VLAN（Virtual LocalArea Network）的中文名为"虚拟局域网"。VLAN 是一种将局域网(LAN)设备从逻辑上划分成一个个网段，从而实现虚拟工作组的数据交换技术。



端口的分隔。物理的交换机可以当作多个逻辑的交换机使用。

网络的安全。不同 VLAN 不能直接通信，杜绝了广播信息的不安全性。

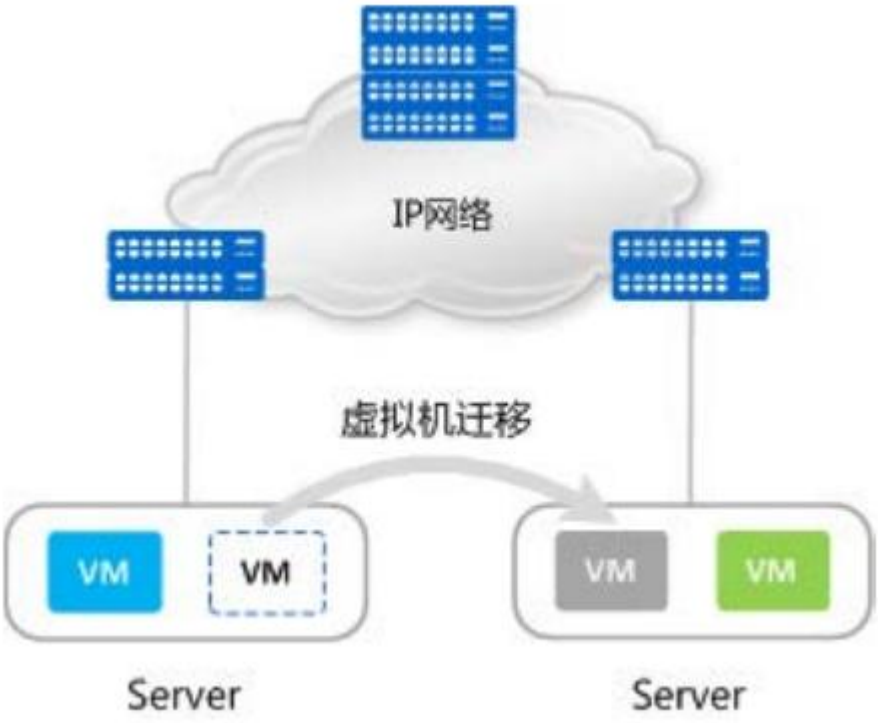
灵活的管理。更改用户所属的网络只更改软件配置

4.4 云计算虚拟化-网络面临挑战

虚拟机规模受网络设备表项规格的限制

传统网络的隔离能力有限

虚拟机迁移 范围 受限



4.5 虚拟化业务下业务隔离

定义：

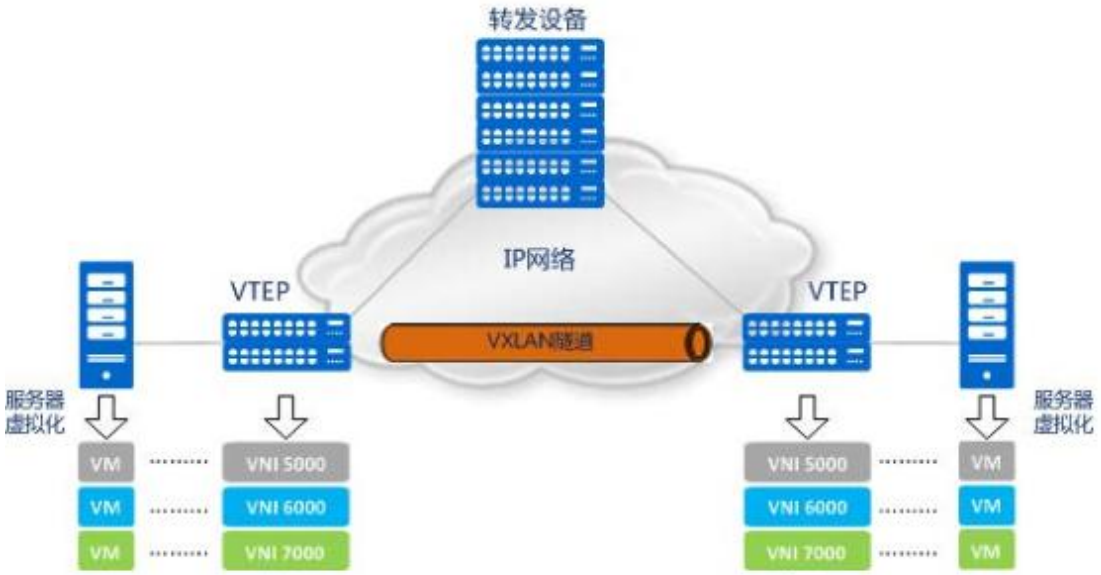
VxLAN（Virtual eXtensible LAN 可扩展虚拟局域网）是基于 IP 网络，采用“MAC in UDP”封装形式的二层 VPN 技术

优点：

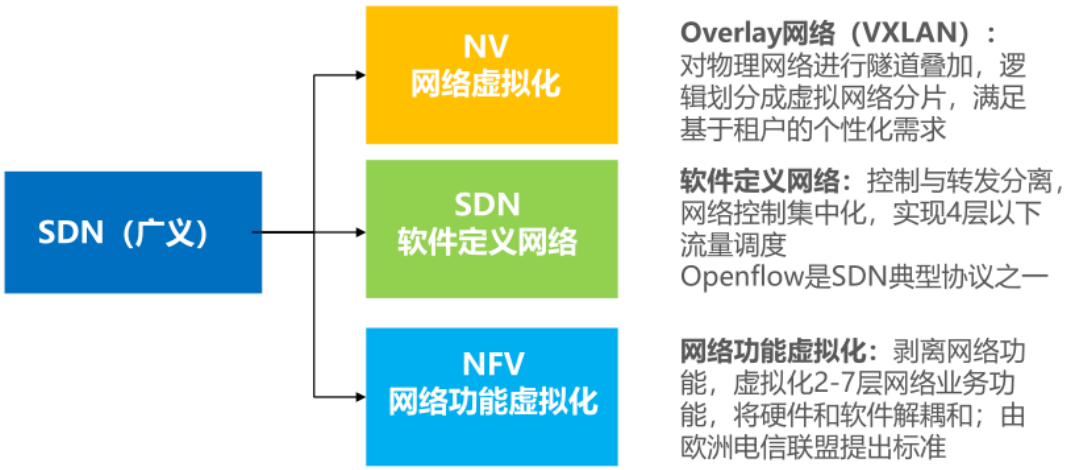
解决虚拟机规模受网络设备表项规格的限制

VxLAN ID 24bit，支持 1600 万个逻辑网络（突破 VLAN 4K 的关键扩展）

通过 VTEP 封装和 VXLAN 隧道技术，“大二层网络” 突破物理上限制



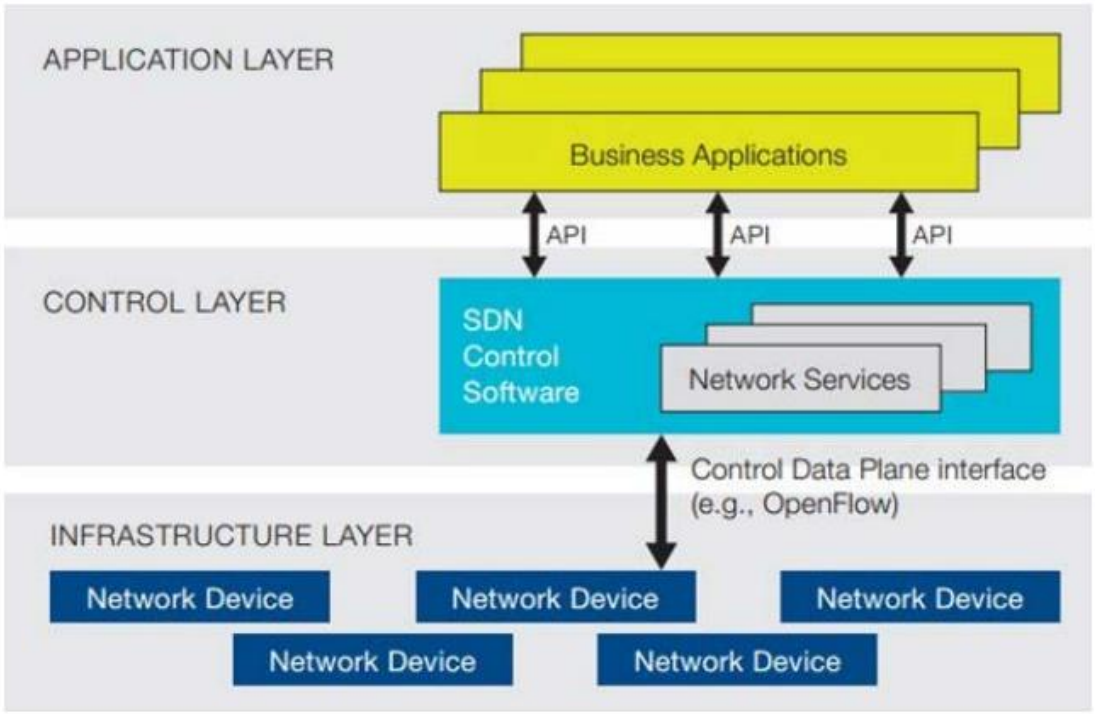
4.6 虚拟化网络-广义 SDN



4.7 SDN 体系架构

核心思想：

OpenFlow 交换机基于流进行转发。同时，传统的控制层面从转发设备中剥离出来，“迁移”到集中控制器上。



4.8 虚拟网络设备

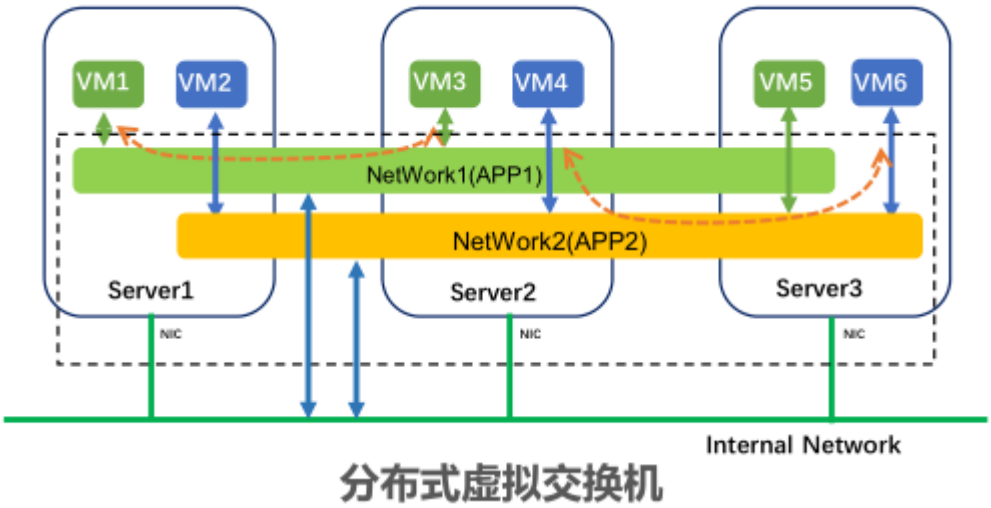
分布式虚拟交换机

简化管理，自动同步主机的交换网络配置

支持 VLAN/VXLAN.

在主机之间迁移虚拟机时，网络配置自动跟随

为第三方提供接口



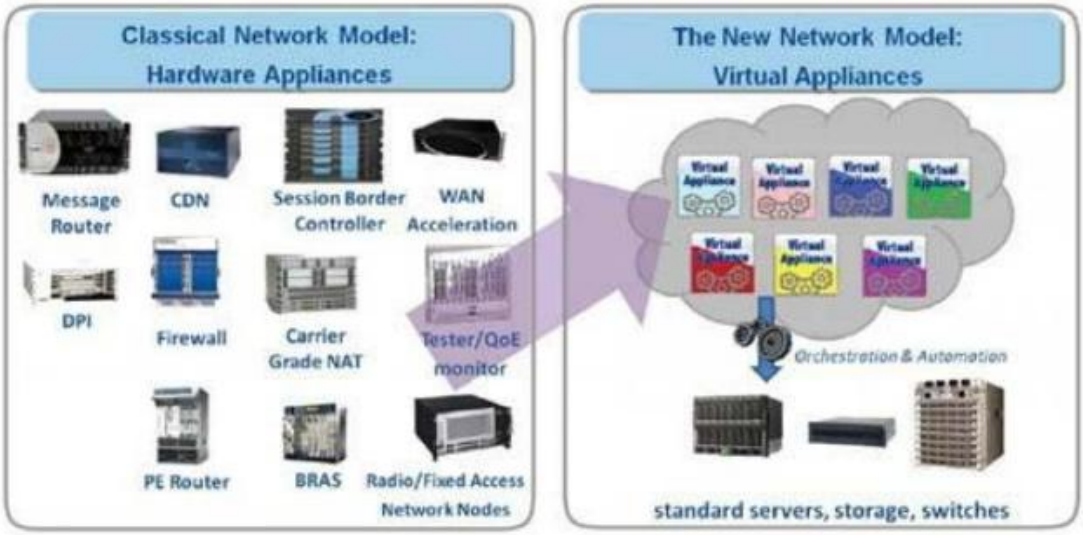
分布式虚拟交换机

分布式虚拟路由

4.9 NFV-网络功能虚拟化

将现有的各类网络设备功能整合进标准的 X86 服务器上。

网络功能虚拟化（Network Functions Virtualization,NFV）是一种对于网络架构（Network Architecture）的概念，它利用虚拟化技术，将网络节点各阶层（交换、路由、优化、安全、…）的功能，分割成多个功能模块，分别以软件方式实现，不再拘限于硬件架构



4.10 SDN 和 NFV 关系

5 数据中心安全

数据安全 ：

为数据处理系统建立技术和管理的安全保护手段，以确保数据的安全性，可靠性
例如：备份，容灾

网络安全 ：

是指网络系统的硬件、软件及其系统中的数据受到保护，不因偶然的或者恶意的原因而遭受到破坏、更改、泄露，系统连续可靠正常地运行

例如：防火墙，UTM