文档密级：公开

InCloud Sphere 6.12.0

技术白皮书

济南浪潮数据技术有限公司

2024 年 4 月

声明

版权声明

安全声明

公司产品不会主动获取或使用用户的个人数据，仅在您同意使用特定功能或服务时，在业务运营或故障定位的过程中可能会获取或使用用户的某些个人数据（如告警邮件接收地址、IP 地址），公司产品在涉及个人数据的收集、存储、使用、传输、删除等全生命周期的处理活动中，已在产品功能上部署了必要的安全保护措施，同时，您也有义务根据所适用国家或地区的法律法规制定必要的用户隐私政策并采取足够的措施以确保用户的个人数据受到充分的保护。

济南浪潮数据技术有限公司高度重视产品数据安全，公司产品在涉及系统运行和安全数据的全生命周期处理活动中，已严格按照相关法律法规及监管要求，在产品功能上部署了必要的安全保护措施。作为系统运行和安全数据处理者，您有义务根据所适用国家或地区的法律法规制定必要的数据安全政策并采取足够的措施以确保系统运行和安全数据受到充分的保护。

特别提示

您购买的产品、服务或特性等应受济南浪潮数据技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买

或使用范围之内。除非合同另有约定，济南浪潮数据技术有限公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新，如有变更，恕不另行通知。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保，济南浪潮数据技术有限公司不对本文档中的遗漏、变更及错误所导致的损失和损害承担任何责任。

联系我们

售前咨询热线：400-8606708

售后服务热线：400-8600011

浪潮数据官网：https://www.inspur.com

目录

1 前言

浪潮是中国领先的云计算、大数据服务商，已经形成涵盖 IaaS、PaaS、SaaS 三个层面的整体解决方案服务能力，凭借浪潮高端服务器、海量存储、云操作系统、信息安全技术为客户打造领先的云计算基础架构平台，基于浪潮政务、企业、行业信息化软件、终端产品和解决方案，全面支撑智慧政府、企业云、垂直行业云建设。

浪潮 InCloud Sphere 是云计算基础平台的核心组成部分，通过虚拟化技术将服务器、存储、网络等硬件设备资源池化，使整个 IT 环境比单独的物理硬件具有更高的可用性、安全性和扩展性，满足企业对于降低成本、简化管理、提高安全性和扩展性的需求，助力企业核心业务向云计算迁移、构建企业云数据中心。

该技术白皮书介绍了虚拟化技术的基本原理，InCloud Sphere 对计算、存储和网络虚拟化技术的创新和应用，在负载均衡、监控、备份、升级多个方面的自动化运维能力，并描述了几种虚拟化系统的典型应用场景。

2 产品概述

通过阅读本章，您可以了解到：

- InCloud Sphere 产品简介与特点描述
- InCloud Sphere 产品的系统架构

2.1 产品介绍

InCloud Sphere 是浪潮推出的一套企业级开放性虚拟化解决平台。针对传统数据中心的基础架构利用率低、物理基础架构成本日益攀升、IT 管理成本不断提高以及对关键应用故障和灾难保护不足等问题而设计和实现。可以将静态、复杂的 IT 环境转变为动态、易于管理的虚拟数据中心，从而降低数据中心成本。同时，它可以提供先进的管理功能，实现虚拟数据中心的集成和自动化，简化运维，降低管理成本，最终帮助用户把更多的时间和成本转移到对业务的投入上。

InCloud Sphere 基于开源 KVM 设计，重点提升虚拟化系统的可靠性、可用性和安全性。InCloud Sphere 重点聚焦企业核心业务的需求，满足 ERP、CRM、核心数据库、Web、电子商务、多应用整合等典型的关键业务应用，加快推动业务和应用的云化。

客户使用 InCloud Sphere 产品可以创建高性能、高可用、可扩展、可管理、灵活的虚拟服务器基础架构，InCloud Sphere 的主要特性包括：

- 高性能

基于开源 KVM 设计,KVM 可以在一套物理硬件上安全的运行多个虚拟机，作为 Linux 内核的一部分，KVM 以高性能和占用资源少著称；虚拟机的 CPU、内存和 I/O 设备等通过 CPU 绑定、EPT、Pass-through 和半虚拟化驱动等技术进一步增强虚拟机性能。

- 低成本

在计算虚拟化上，通过 CPU、内存的虚拟化技术，实现多个虚拟机在物理机上的同时稳定运行，减少物理机采购数量，降低内存等硬件采购成本。在存储虚拟化上，采用存储的快速克隆、存储精简置备技术，减少对虚拟磁盘的过度调配，可节省或延迟存储设备采购时间，降低硬盘等硬件采购成本。

- 高可用

InCloud Sphere 的虚拟机计算迁移和整机迁移功能可以显著提高虚拟机在资源池内的生命力，在不关机的情况下进行运维工作，对企业应用无任何影响；当 Hypervisor 层或服务器层发生故障时，InCloud Sphere 均可自动重启服务器承载虚拟机，或为虚拟机重新选择健康的服务器启动，保护所有的虚拟化应用。支持虚拟 CPU、虚拟内存、虚拟磁盘、虚拟网卡的热添加功能，减少系统计划内宕机时间，并为企业带来更高的可用性。

- 管理控制台

提供 B/S 架构的 InCloud Sphere iCenter 管理控制台，方便企业灵活部署；iCenter 可以通过一个界面即可提供所有的虚拟机监控、常规管理和命令行界面管理功能。管理员可以在任意系统通过浏览器打开 B/S 架构的管理控制台，轻松管理整个虚拟化环境。同时，InCloud Sphere 提供了轻量级的 C/S 架构管理控制台，该控制台专注于虚拟机的管理，提供高分辨率的虚拟机桌面以及文件在虚拟机和用户登陆终端之间的便捷复制功能。

- 扩展性

提供完整的 REST API 和软件开发工具包 SDK，完全兼容 OpenStack，并可以根据不同客户需求进行接口的扩展和开发。

2.2 系统架构

InCloud Sphere 的体系架构如图 2.2-1 所示：

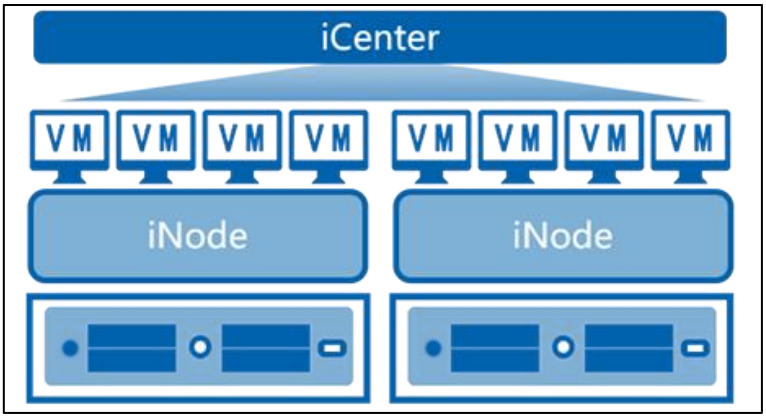


图 2.2-1 InCloud Sphere 逻辑架构

InCloud Sphere 逻辑架构中的组件详细介绍如下：

- iCenter 是 InCloud Sphere 的管理节点，它能够监控并管理 InCloud Sphere 基础设施，包含 iNode 主机、虚拟机和其他物理/虚拟资源。同时，iCenter 为系统管理员提供了一套图形化的系统管理界面。
- iNode 是 InCloud Sphere 的计算节点，可采用物理机或虚拟机的部署模式，将硬件层抽象虚拟化，提供虚拟机运行所需的 CPU、内存、存储和网络资源，同时控制虚拟机的运行。
- VM 是 Virtual Machine 的简称，即虚拟机。VM 是运行在 InCloud Sphere 上的完整计算机软硬件系统，能够提供与传统物理机相同的功能和性能。

3 技术原理

通过阅读本章，您可以了解到：

- InCloud Sphere 系统的设计实现方法
- InCloud Sphere 所采用的经典虚拟化技术和原理

3.1 系统设计

InCloud Sphere 的系统架构如图 3.1-1 所示：

图 3.1-1 InCloud Sphere 系统架构

对图 3.1-1 InCloud Sphere 系统架构中的不同组件详细介绍如下：

- 计算虚拟化通过对 KVM、libvirt、qemu 和 seabios 等开源组件的深度适配与性能优化，增强计算虚拟化能力，并对计算服务进行封装；
- 存储虚拟化实现了对本地、NFS、CFS 和 LVM 等多种存储资源的池化和统一管理，增强存储虚拟化能力，并对存储服务进行封装；
- 网络虚拟化基于 OpenvSwitch 等技术实现了网络功能虚拟化，并提供了 SR-IOV 交换机、Macvtap 交换机等高速网络功能；
- 运维层在计算虚拟化、存储虚拟化和软件虚拟化基础上，实现了访问控

制、资源监控、资源调度、高可用（HA）等功能，增强了 InCloud Sphere 的自动化运维能力。

- 服务层为系统管理员提供了友好的图形管理界面，为开发人员提供了完备的 REST API 和 Java SDK 接口，无缝对接上层云管理平台（CMP）和 OpenStack。

3.2 虚拟化核心技术与原理

虚拟化 (Virtualization) 是资源的逻辑表示，实现虚拟资源与物理硬件解耦。虚拟化技术的实现形式是在系统中加入一个虚拟化层，将下层的资源抽象成另一形式的资源，提供给上层使用。从广义上来说，虚拟化就是一种采用软硬件分区、聚合、部分或完全模拟、分时复用等方法来管理计算资源、构造一个或者多个计算环境的技术。所构建的计算环境还需要有以下三个特点：

- 保真性：应用程序在虚拟机上执行，将表现为与在物理硬件上相同的执行行为。
- 高性能：在虚拟执行环境中，应用程序的绝大多数指令能够在 VMM 不干预的情况下，直接在物理硬件上执行。
- 安全性：物理硬件应该由 VMM 全权管理，被虚拟出来的执行环境中的程序（包括操作系统）不得直接访问硬件。

在操作系统与硬件之间，虚拟化软件通过空间上的分割、时间上的分时以及模拟，抽象出一个虚拟的硬件接口，向上层操作系统提供一个与它原先期待一致的服务器硬件环境，使得上层操作系统可以直接运行在虚拟环境上，可允许多个操作系统同时运行在单个物理服务器上。

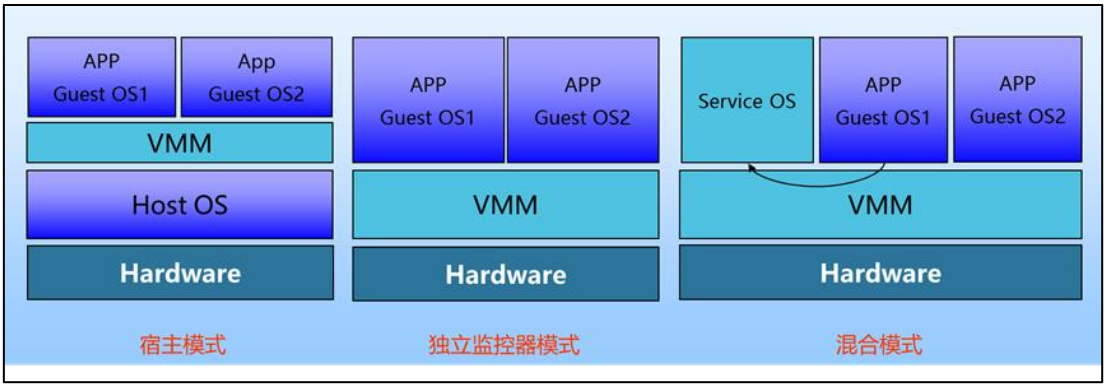


图 3.2-1 虚拟机监控器类型

服务器虚拟化的虚拟化软件层称为虚拟机监控器 (Virtual Machine Monitor, VMM)，也称为 Hypervisor，如图 3.2-1 所示，常见虚拟机监控器分为三类：

- 宿主模式：在 VMM 之下还有一层宿主操作系统，并不直接运作在裸机上。由于 Guest OS 对硬件的访问必须经过宿主操作系统，因而带来了额外的性能开销，但可充分利用宿主操作系统提供的设备驱动和底层服务来进行内存管理、进程调度和资源管理等。
- 独立监控器模式：VMM 直接运作在裸机上，管理和使用底层硬件资源，Guest OS 对硬件资源的访问都要通过 VMM 来完成，作为底层硬件的直接操作者，VMM 拥有硬件的驱动程序。
- 混合模式：为宿主模式和独立监控器模式的综合，VMM 直接运行在裸机上，但是驱动程序需要由 Service OS 提供，Service OS 为运行在 VMM 上的一台特殊虚拟机。

虚拟化技术的实质：将底层资源进行分区，并向上层提供特定的和多样化的执行环境。

虚拟机的定义：是指在一个硬件平台上模拟多个高效的、独立的和实际硬件相同的虚拟硬件系统，在每个虚拟硬件系统上都可以运行不同的客户操作系统

(Guest OS)。Guest OS 通过虚拟机监控器访问实际的物理资源。因此操作系统和应用程序在虚拟机中的运行方式与它们在物理服务器上的运行方式没有区别。

虚拟机和物理机的本质区别：与物理服务器相比，虚拟机不是由真实的电子元件组成，而是由一组虚拟组件（文件）组成，这些虚拟组件与物理服务器的硬件配置无关，与物理服务器相比，虚拟机具有以下优势：

- **抽象解耦**：可以在任何 X86 架构的服务器上运行。虚拟机的运行环境不再受限于具体硬件配置，可以进行灵活的配置和迁移。
- **分区隔离**：多个具有独立操作系统的虚拟机同时运行，虚拟化层为其划分服务器资源，虚拟机之间在数据处理、网络连接和数据存储等方面实现安全隔离。
- **封装移动**：可以将整个虚拟机系统（包括虚拟硬件、操作系统和配置好的应用程序）封装于一个或多个文件之中，通过简单的文件复制实现快速部署、备份及还原，同时还可以在不同的物理服务器之间进行迁移，甚至可以在虚拟机正在运行的情况下进行迁移。
- **弹性扩展**：可以对单个物理服务器上的虚拟资源（虚拟 CPU、虚拟网卡、虚拟磁盘等）在不关闭虚拟机的情况下，进行按需动态扩展。

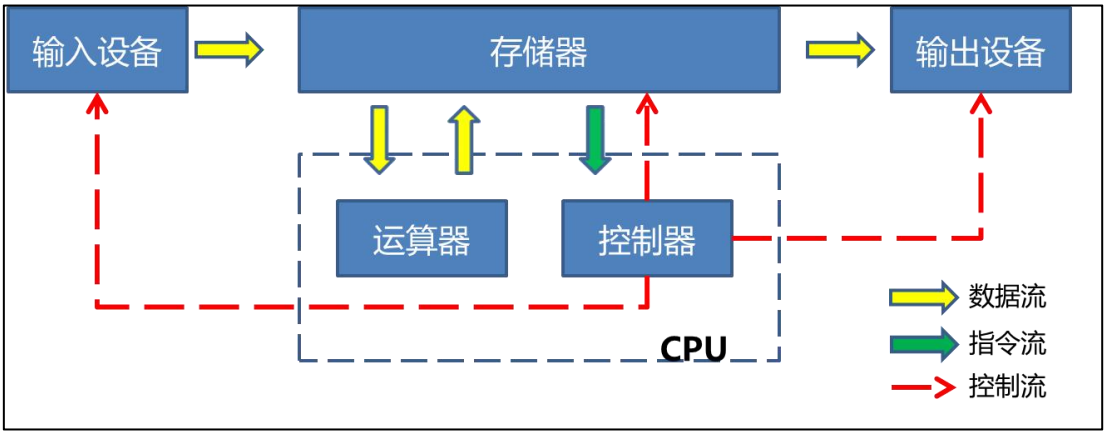


图 3.2-2 冯•诺依曼体系结构

如图 3.2-2 所示，冯•诺依曼体系结构中计算机被划分为 CPU（运算器和控制器）、存储器和输入输出设备三个模块，相应的 VMM 对物理资源的虚拟可以划分为三个部分：CPU 虚拟化、内存虚拟化和 I/O 设备虚拟化，其中以 CPU 虚拟化最为关键。

3.2.1 CPU 虚拟化

VMM (Virtual Machine Monitor)对物理资源的虚拟可以划分为三个部分：CPU 虚拟化、内存虚拟化和 I/O 设备虚拟化。

3.2.1.1 经典 CPU 虚拟化方法

现代计算机体系结构一般至少有两个特权级（即用户态和核心态，X86 架构的 CPU 有四个特权级 Ring0~ Ring3）用来分隔系统软件和应用软件。那些只能在处理器的最高特权级（内核态）执行的指令称之为特权指令，一般可读写系统关键资源的指令（即敏感指令）绝大多数都是特权指令（X86 存在若干敏感指令是非特权指令的情况）。如果执行特权指令时处理器的状态不在内核态，通常会引发一个异常而交由系统软件来处理这个非法访问（陷入）。经典的虚拟化方法就是使用“特权解除”和“陷入-模拟”的方式，即将 Guest OS 运行在非特权级，而将 VMM 运行于最高特权级（完全控制系统资源）。解除了 Guest OS 的特权级后，Guest OS 的大部分指令仍可以在硬件上直接运行，只有执行到特权指令时，才会陷入到 VMM 模拟执行（陷入-模拟）。“陷入-模拟” 的本质是保证可能影响 VMM 正确运行的指令由 VMM 模拟执行，大部分的非敏感指令还是照常运行。

3.2.1.2 X86 架构虚拟化方法

X86 体系结构拥有四个特权级以分隔系统软件和应用软件，如图 3.2.1-1 (a)所示。其中，内核在 Ring 0 级，应用程序在 Ring 3 级。依照经典的 CPU 虚拟化技术，在引入虚拟化之后，在全虚拟化和半虚拟化场景中，VMM 接管系统最

高权限，由此 VMM 在 Ring 0 级，内核迁移至 Ring 1 级，如图 3.2.1-1 (b)所示。

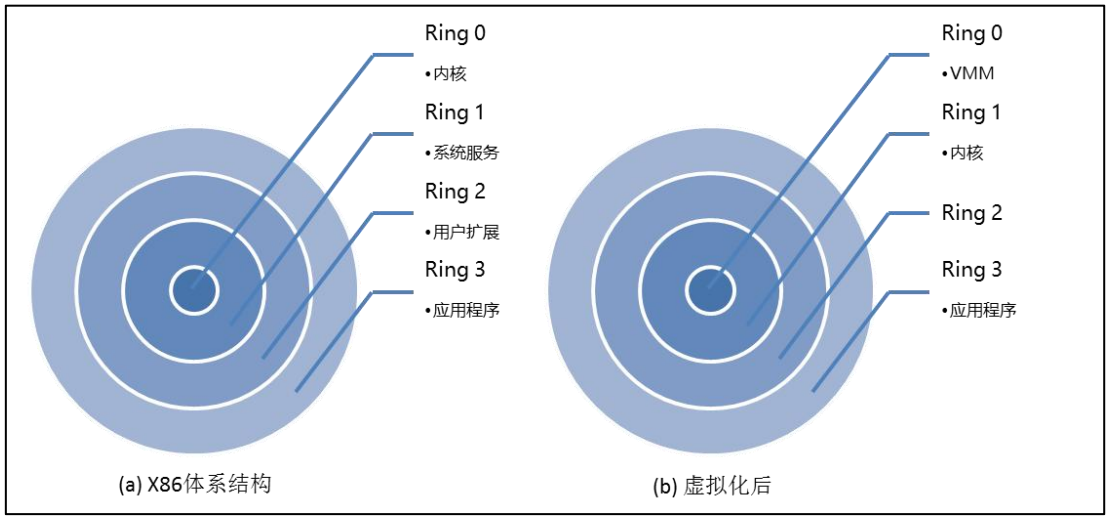


图 3.2.1-1 特权级

在虚拟化技术中，具有以下功能的指令被认为是敏感指令：操作特权资源、修改虚拟机的运行模式、修改物理机的运行状态、读写敏感的寄存器和内存、访问存储保护系统、内存系统或地址重定位系统以及所有 I/O 指令。但是 X86 架构中存在若干敏感指令是非特权指令的情况，即敏感指令中只有一部分属于特权指令，如图 3.2.1-2 所示在传统的计算机体系架构中，只有特权指令才能在最高特权级运行。由此，虚拟机调用了某些会影响正常运行的敏感指令便不会陷入到 Ring 0 级，而是由 VMM 模拟执行，从而产生“虚拟化漏洞”。

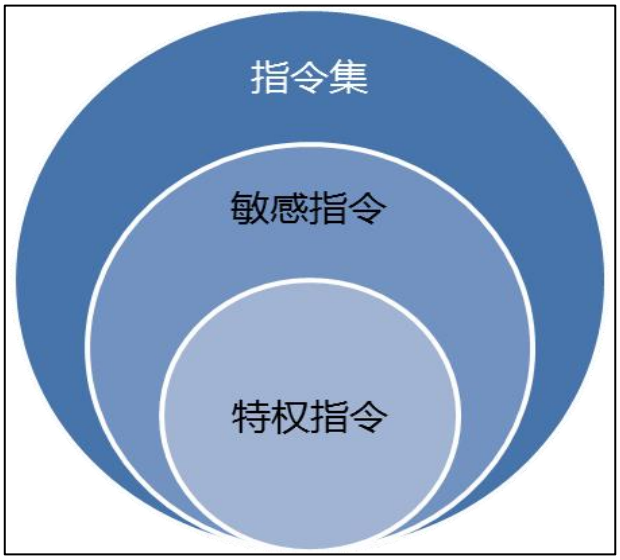


图 3.2.1-2 虚拟化漏洞

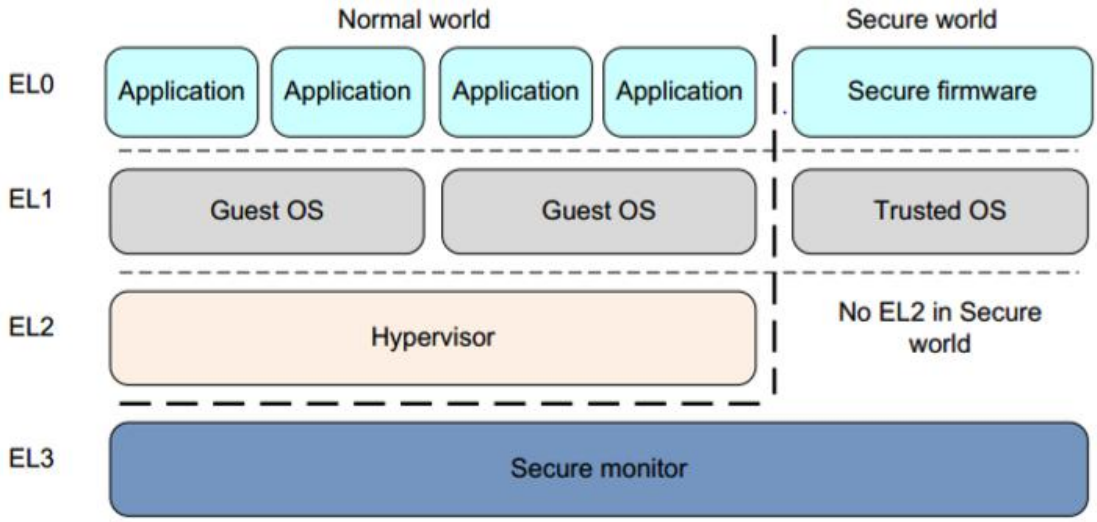
为解决以上“虚拟化漏洞”问题，全虚拟化技术在执行时将 VM 上执行的 Guest OS 指令翻译成 X86 指令集，其某一子集中的敏感指令被替换成陷入指令，陷入到 Ring 0 模式执行；半虚拟化技术通过修改 Guest OS 的代码，将含有敏感指令的操作，替换为对 VMM 的超调用 Hypercall，将控制权转移到 VMM；硬件辅助虚拟化技术将非根模式下所有敏感指令重新定义，使它们能不经虚拟化直接运行或通过“陷入再模拟”的方式处理，在模式切换过程中，上下文的保存恢复由硬件来完成，极大提高了切换效率。

而在硬件辅助虚拟化场景中，其基本思想就是引入新的处理器运行模式和新的指令，使得 VMM 和 Guest OS 运行于不同的模式下，Guest OS 运行于受控模式，原来的一些敏感指令在受控模式下全部会陷入 VMM，这样就解决了部分非特权的敏感指令的“陷入-模拟”难题，而且模式切换时上下文的保存恢复由硬件来完成，这样就大大提高了“陷入-模拟”时上下文切换的效率。以 Intel VT-x 技术为例，该技术增加了在虚拟状态下的两种处理器工作模式，即根模式和非根模式。如图 3.2.1-3 所示，Guest OS 在非根模式的 Ring 0 级，VMM 在根模式下。通过硬件辅助的方式，尽可能少地在客户机和 VMM 之间切换，减少上下文切换的开销，并减少 VMM 的模拟任务。

图 3.2.1-3 硬件辅助虚拟化特权级

3.2.1.3 ARM 架构虚拟化方法

ARMv8-A 架构是 ARM 公司为满足新需求重新设计的一个架构，支持 32 位执行状态与 64 位执行状态，并保持了与 ARMv7 软件的向后兼容性。ARMv8 把之前架构中的 processor mode 的概念去掉（或者说淡化），引入了 Execution State、Exception Level、Security State 等新特性。如下图所示，ARMv8 架构将处理器模式划分成四个异常级别（Exception Level），异常级别决定了特权级别。其中 User Application 位于特权等级最低的 EL0，Guest OS 位于 EL1，提供虚拟化支持的 Hypervisor 位于 EL2，提供 Security 支持的 Security Monitor 位于 EL3。其中 EL3 是 ARM TrustZone 技术的基础，ARMv8-A 提供两种安全状态，使得普通操作系统与受现任操作系统(Trusted OS)在同一硬件上运行，当前 ARM 虚拟化不支持安全模式虚拟化，不再展开介绍。在最新的 ARMv8.4 中，已经支持 secure privilege level 2。将来应该会引入最新版本 ARM CPU 中。



只有在异常(如：中断、page faults 等)发生时（或者异常处理返回时）才能切换 Exception level（这也是 Exception level 的命名原因，为了处理异常）。当异常发生时，有两种选择，停留在当前的 EL，或者跳转到更高的 EL，EL 不能降级。同样，异常处理返回时，也有两种选择，停留在当前的 EL，或者调到更低的 EL。

EL0 => EL1: SVC (system call)

EL1 => EL2: HVC (hypervisor call)

EL2 => EL3: SMC (secure monitor call)

上图中 hypervisor 就是运行在 EL2 模式下，而客户机系统则是运行在 EL1 模式下。

ARMV8 从设计之初就提供了对虚拟化的硬件辅助支持,主要包括以下几点：

一、增加运行在 Non-secure privilege level 2 的 Hypervisor 模式，Guest OS kernel 执行在 EL1，userspace 执行在 EL0。

二、使大部分的敏感指令可以本地执行(native-run)，在 EL1 上而不必 trap 及 emulation，而仍需要 trap 的敏感指令会被 trap 到 EL2 (hypervisor mode HYP)。在 Guest OS 中执行下面的指令陷入到 EL2：

1、访问虚拟机内存控制寄存器，如 TTBRn and TTBCR 系统指令，如 cache 和 TLB 操作指令；

2、访问辅助控制寄存器 ACTLR_EL1；

3、读取 ID 寄存器；

4、执行 WFE 或者 WFI 指令；

三、在 EL2 中实现额外的存储器转换层，称为“阶段 2 转换”。hypervisor 将为每个虚拟机创建和管理 Stage 2 的页表。

四、提供 3 种虚拟异常的支持，Virtual SError、Virtual IRQ 和 Virtual FIQ。物理异常被配置为 hypervisor 捕获，虚拟异常将会被注入到 Guest OS

3.2.2 内存虚拟化

操作系统对内存有两点认识，首先，内存都是从物理地址 0 开始的；其次，内存是连续的，或者至少在一些大的粒度（如 256M）上连续。但是在虚拟化环境下，物理内存被多个客户机操作系统同时使用，物理地址 0 只有一个，可以保证为每台虚拟机分配连续内存，却牺牲了使用效率。而 VMM 需要“欺骗”客户机操作系统，使客户机操作系统认为其对内存的两点要求都还满足。这种欺骗的过程，即内存虚拟化。

内存虚拟化的核心，即引入一层新的地址空间：客户机物理地址空间，如图 3.2.2-1 所示：

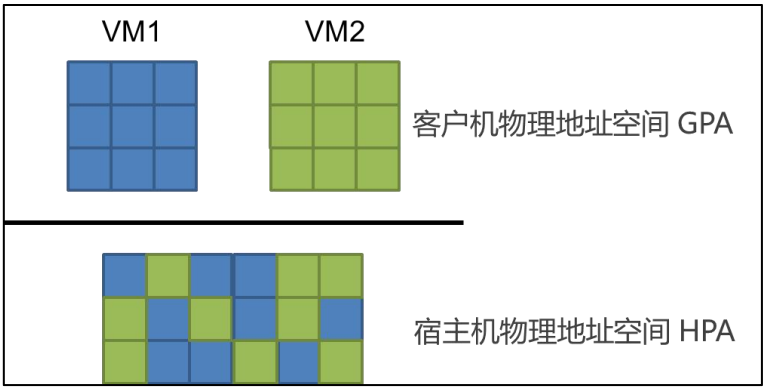


图 3.2.2-1 客户机物理空间地址

由此，内存虚拟化前两层地址映射关系扩展为三层地址映射关系，如下图所示：

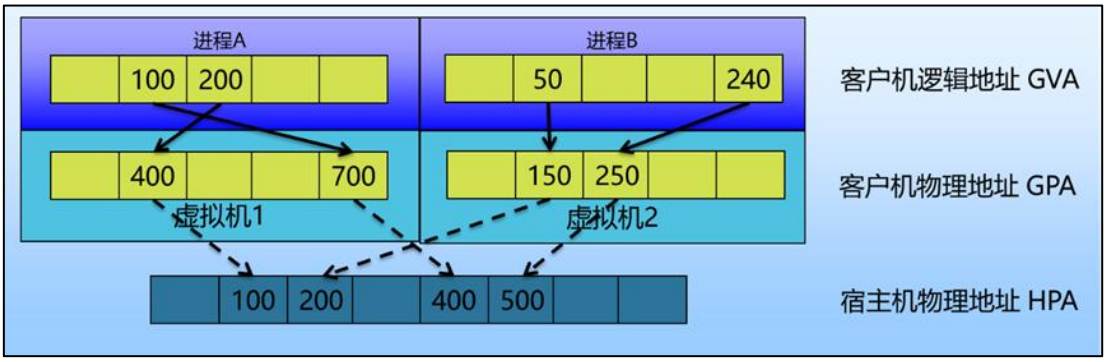


图 3.2.2-2 内存地址空间三层映射

内存虚拟化要处理好以下两个问题：首先，给定一个虚拟机，维护客户机物理地址到宿主机物理地址之间的映射关系；其次，截获虚拟机对客户机物理地址的访问，并根据所记录的映射关系，将其转换至宿主机物理地址。

在全虚拟化技术中，VMM 负责了客户机物理地址到宿主机物理地址间的映射；半虚拟化技术中，KVM 将客户机物理地址到机器地址的翻译表（P2M 表）暴露给客户操作系统，从 VMM 中获得属于本虚拟机的内存空间信息；在硬件辅助虚拟化技术中，以 Intel-EPT 技术为例，通过硬件实现的扩展页表（Extended Page Table）技术，加速实现了客户机逻辑地址到宿主机物理地址的转换，如下图所示。

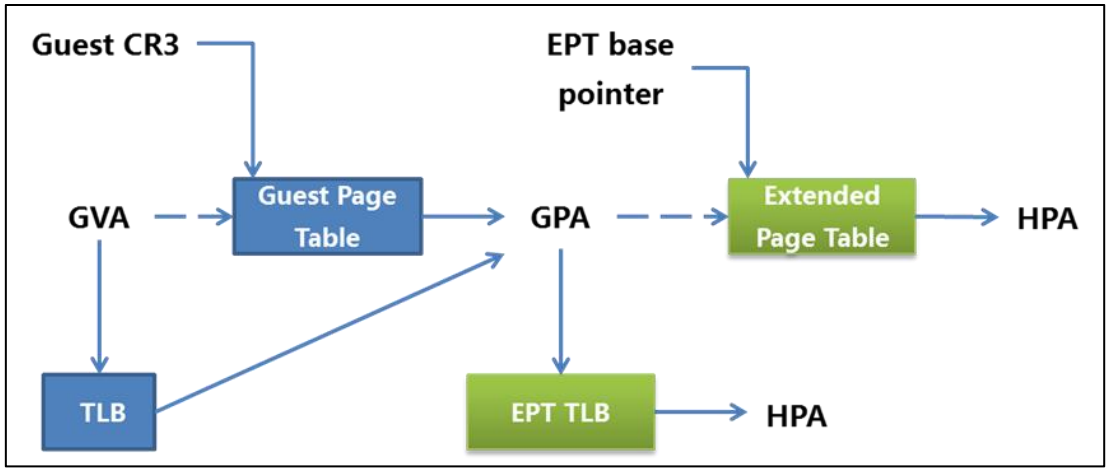


图 3.2.2- 3 EPT 扩展页表

3.2.3 I/O 设备虚拟化

在虚拟环境中，I/O 设备面临着有限的外设资源与多个客户操作系统对外设访问需求的矛盾。而 VMM 则需要截获客户机操作系统对外设的访问请求，以软件或硬件辅助的方式模拟真实物理设备的效果。

设备模拟的方式上，全虚拟化技术使用软件精确模拟与物理设备完全一样的接口，客户操作系统驱动无须修改就能驱动这个虚拟设备；半虚拟化技术通过修改客户操作系统，使前后端相互协作，提供更加高效的 I/O 虚拟化；而硬件辅助虚拟化技术直接将物理设备分配给某个客户操作系统，由客户操作系统直接访问 I/O 设备不经过 VMM。

VMM 通过 I/O 虚拟化来复用有限的外设资源，其通过截获 Guest OS 对 I/O 设备的访问请求，然后通过软件模拟真实的硬件，目前 I/O 设备的虚拟化方式主要有三种：

1、设备接口完全模拟：

即软件精确模拟与物理设备完全一样的接口，Guest OS 驱动无须修改就能驱动这个虚拟设备，VMware 即使用该方法。

优点：没有额外的硬件开销，可重用现有驱动程序；

缺点：为完成一次操作要涉及到多个寄存器的操作，使得 VMM 要截获每个寄存器访问并进行相应的模拟，这就导致多次上下文切换；由于是软件模拟，性能较低。

2、前端／后端模拟：

VMM 提供一个简化的驱动程序（后端，Back-End），Guest OS 中的驱动程序为前端 (Front-End, FE)，前端驱动将来自其他模块的请求通过与 Guest OS 间的特殊通信机制直接发送给 Guest OS 的后端驱动，后端驱动在处理完请求后再发回通知给前端，KVM 即采用该方法。

优点：基于事务的通信机制，能在很大程度上减少上下文切换开销，没有额外的硬件开销；

缺点：需要 VMM 实现前端驱动，后端驱动可能成为瓶颈。

3、直接映射：

即直接将物理设备分配给某个 Guest OS，由 Guest OS 直接访问 I/O 设备（不经 VMM），目前与此相关的技术有 IOMMU（Intel VT-d、SR-IOV 等），旨在建立高效的 I/O 虚拟化直通道。

优点：可重用已有驱动，直接访问减少了虚拟化开销；

缺点：需要购买较多额外的硬件。

4 虚拟化能力

通过阅读本章，您可以了解到：

- InCloud Sphere 系统的主要功能
- InCloud Sphere 各功能的实现原理

4.1 虚拟机

4.1.1 CPU 虚拟化

4.1.1.1 CPU 绑定

通过 CPU 绑定选项，可以精确控制虚拟机 CPU 在主机的物理核心之间分布的方式。使用 CPU 绑定可以向虚拟机分配特定处理器，通过此操作，可以将虚拟机只分配给多处理器系统中特定的可用处理器。

4.1.1.2 CPU 热插拔

当用户应用负载增长超过原规划导致虚拟机 CPU 资源不足时，可通过 CPU 热插技术为虚拟机增加 CPU 资源，在此过程中虚拟机不需要关机，从而不会影响用户业务的运行；相应的，若分配给虚拟机的资源超过其实际需要的份额时，可以通过 CPU 热拔技术将部分 CPU 资源释放给虚拟化系统，以实现资源的更优配置。

4.1.1.3 CPU QoS

CPU QoS 只在各虚拟机竞争计算资源的时候才发挥作用，如果没有竞争情况发生，有需求的虚拟机可以独占物理 CPU 资源。CPU 份额是在多个虚拟机竞争物理 CPU 的时候按比例分配计算资源。以一个主频为 2.8GHz 的单核物理机为例，如果上面运行有三台单核的虚拟机。三个虚拟机 A,B,C，它们的份额分别设置为低，正常，高。当三个虚拟机内部都运行满 CPU 负载的应用时，Hypervisor 会根据三个虚拟机的份额按比例分配计算资源，份额为低的虚拟机 A 的计算能

力约为 400MHz，份额为正常的虚拟机；B 获得的计算能力约为 800MHz，份额为高的虚拟机；C 获得的计算能力约为 1600MHz。（以上举例仅为说明 CPU 份额的概念，实际应用过程中情况会更复杂）。CPU 份额可以控制一个服务器上的各个虚拟机在竞争 CPU 资源时的资源占用率，以便针对不同等级的用户提供服务。

4.1.1.4 CPU 限制

控制虚拟机占用主机物理 CPU 资源的上限。值可以设置为指定数值，或者设置为不限。CPU 限制值设置的越大，虚拟机占用的主机物理资源就越多。

4.1.1.5 CPU 预留

InCloud Sphere 虚拟化平台为虚拟机提供 CPU 资源预留的功能，该功能可以给虚拟机指定一定的 CPU 资源（单位 MHz），该虚拟机运行期间，宿主机进程和其他虚拟机均无法使用该部分 CPU 资源，只有预留了这部分 CPU 资源的虚拟机方可使用；该功能能够确保一些关键虚拟机在主机 CPU 资源紧张的情况下依然能够使用足够的 CPU 资源，以保持正常运行。

CPU 预留步骤如下：

1.虚拟机在创建或关机状态编辑虚拟机时，可设置 CPU 资源预留；

2.预留单位为 MHz，最大预留量为虚拟机 CPU 数量*主机 CPU 频率；

3.虚拟机开机后主机便会预留相应的 CPU 资源；

4.虚拟机关机后主机会释放预留的 CPU 资源，以减轻主机 CPU 压力。

4.1.2 内存虚拟化

4.1.2.1 内存热添加

InCloud Sphere 对 VM 内存管理增加了内存热添加的功能，如果在创建 VM 时分配固定数量的内存资源不能达到 VM 的要求，可以通过内存热添加功能来

对内存资源进行动态的获取。要增大 InCloud Sphere 环境中物理内存的利用率，可以通过界面开启内存热添加功能来进行，这是一种能够在 VM 运行期间不中断业务便可以动态增加内存的管理功能。

4.1.2.2 内存预留

InCloud Sphere 内存预留功能是指预留一部分主机物理内存供 VM 使用。一旦 VM 内存需求达到了预留的规定值，这部分内存就完全分配给该 VM，不可以再分配出去（其他任务或者虚拟机）。如果 VM 上运行关键的应用，可以考虑为虚拟机开启内存预留功能。简单的做法是设置和最大内存一样大小。但这可能造成不必要的资源浪费。内存预留保证了该 VM 的一部分物理内存需求，但意味着整个主机可以共享的物理内存减少了。可能影响到能够运行在该主机上的 VM 的个数。推荐按照虚拟机实际活动内存的需求进行配置。

4.1.2.3 内存气泡

InCloud Sphere 支持内存气泡功能。通过 Ballooning 动态调整虚拟机内存，主机内存按需分配。实现原理通过虚拟机中申请内存，然后将申请的内存通知给 qemu，然后 qemu 侧再将内存释放掉，可以让其他虚拟机使用，达到内存复用的能力。

内存气泡功能需要添加 Virtio-Balloon 后端设备，并且要在虚拟机内部安装 Balloon 驱动。

4.1.2.4 NUMA 调度

InCloud Sphere 支持根据宿主机 NUMA 节点的资源使用情况，选择合适的 NUMA 节点放置虚拟机，并使虚拟机的 CPU 资源和内存资源在同一个 NUMA 节点上，保证虚拟机访问本地物理内存，减少了内存访问延迟，可以提升 VM 性能。

InCloud Sphere 支持 NUMA 架构虚拟机（vNUMA），为虚拟机的 NUMA

节点配置 CPU 和内存拓扑结构，保证 vNUMA 节点的本地性。同时，通过对应虚拟机 vNUMA 和宿主机 NUMA 拓扑结构，使虚拟机能够充分利用整台宿主机的物理资源，构建巨型虚拟机。

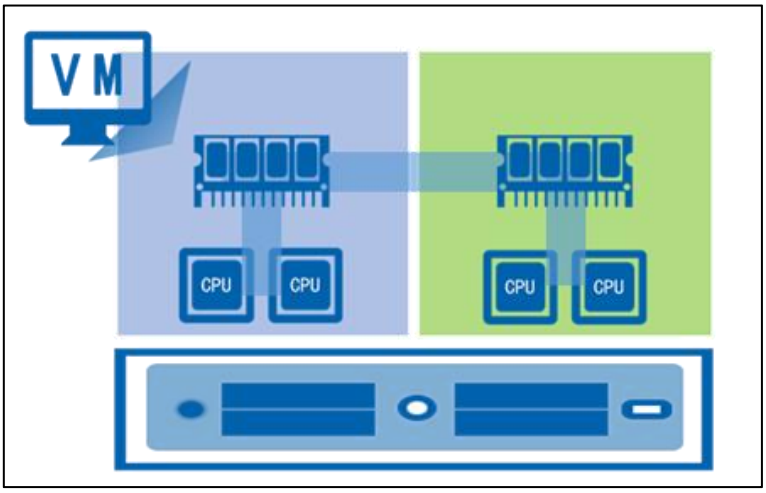


图 4.1.2-1 虚拟机 NUMA 节点调度

4.1.2.5 大页内存

大页内存（hugepage）来取代传统的 4kb 内存页面，同等物理内存下使得管理虚拟地址页表数变少，加快了从虚拟地址到物理地址的映射（TLB 缓存）并通过禁用内存页面的 swap in/swap out 等手段综合提高了内存的整体性能。对于一些需要较大内存以及对内存性能有较高要求的应用尤为重要。

根据大页内存每页占据的大小，可以再分为不同规格大小大页内存。以 Intel 架构下目前支持的大页内存为例，其当前支持 2M 与 1G 的大页内存，支持的差异取决于 cpu 的 flag 参数，如下分别表示支持的大页类型：

```
#cat   /proc/cpuinfo|grep -w pse     //2M
#cat   /proc/cpuinfo|grep -w dpe1gb   //1G
```

2M 与 1G 大页内存使用方式略有不同。大页内存为物理内存连续的一段内存，随着系统运行时间的增长，内存碎片递增。在分配 1G 大页内存时，如不满足条件，大页内存很可能存在失败情况，2M 分配表现稍好。另外运行时分配 1G

大页内存，要开启 Contiguous Memory Allocator (CONFIG_CMA) 选项。在同时支持 2M 与 1G 模式下，建议优先选用 1G 的大页内存类型可以达到更加性能。由大页内存组成的大页内存池，如果系统支持 2M 与 1G 模式，可以使用一种类型的也可以混合使用内存池。因为大内内存通过 hugtlbfs 文件系统挂载点对外使用接口，因此混合使用时需要挂载多个 hugtlbfs 文件系统。与此同时虚拟化下使用混合内存池，也需要对 qemu 进行配置修改，使用过于复杂。鉴于此 InCloud Sphere 提供简单高效的单类型大页内存给用户。

大页内存一旦申请预留分配，不能再用于其他用途。因此合理分配，既能满足使用需求，也不会造成浪费。大页内存的预留分配需要从以下几个方面综合考虑：

- numa 结构；
- 物理内存布局；
- 大页内存类型；
- 内存预留比；
- 虚拟机内存使用。

现在处理器通常支持 numa 架构，应用程序或者虚拟机访问所在 numa node 的本地内存性能高于访问远端内存性能，这点非常重要。同时物理内存的布局也影响着 numa node 上的大页内存分配情况，大页内存统一分配的情况下，预留的大页内存会均分在所有 numa node 上，因此需要保证 numa node 的物理内存要大于其上准备预留的内存。关于大页内存类型与内存预留比，上文已有表述，总结为优先选用 1G 的大页内存类型可以达到更加性能，合理分配，充分利用。在 InCloud Sphere 中，大页内存的使用对象主要是虚拟机与 dpdk，其中主要是虚拟机。因此根据虚拟机业务情况，分配适量的大页内存也是一个重要的参考方面。

4.1.3 虚拟磁盘

4.1.3.1 厚置备和精简置备

区别于传统的厚置备置零，InCloud Sphere 虚拟磁盘管理提供的厚置备延迟置零类型采取预分配的方式，在创建时为虚拟磁盘分配所需空间但不会擦除物理设备上的数据，仅当有 I/O 操作时才置零，然后完成 I/O，既可以预分配空间，又提高了空间分配效率；

精简置备并未在虚拟机磁盘创建的时候为其分配所要求的全部空间，只在磁盘文件的元数据区的文件大小字段写入了虚拟的的文件大小，而元数据区实际记录磁盘块索引的字段为空，故并未实际分配对应的磁盘块。在虚拟机向磁盘中写入数据的时候，才按需、动态的在物理存储空间中分配对应的存储空间。三种分配方式的对比如图 4.1.3-1 所示。

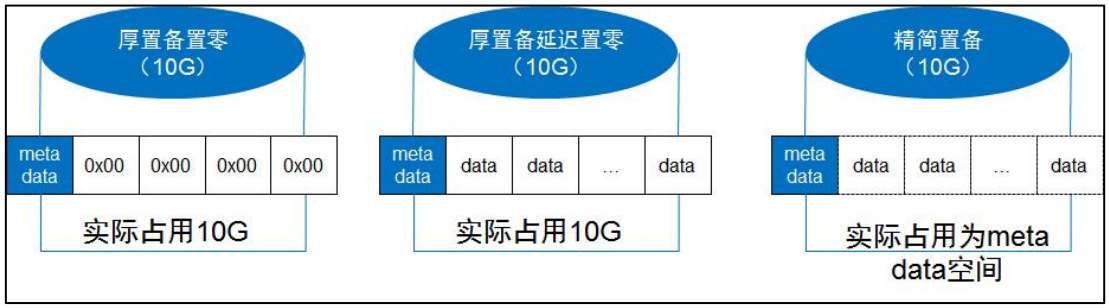


图 4.1.3-1 虚拟磁盘类型

4.1.3.2 磁盘快照

InCloud Sphere 磁盘管理提供快照功能，采用写入时复制技术(Copy on Write)，当某个 block 将被改写的时候，该 block 首先被 copy 到其他地方（即拷贝到快照系统指定的某个位置），然后再在它原来的位置进行改写。这样能保证每个快照文件数据的完整性，各个快照之间是相互独立的，不会因为某一个快照损坏导致无法恢复，可以实现快照快速创建、恢复、删除等功能，便于快照的管理。

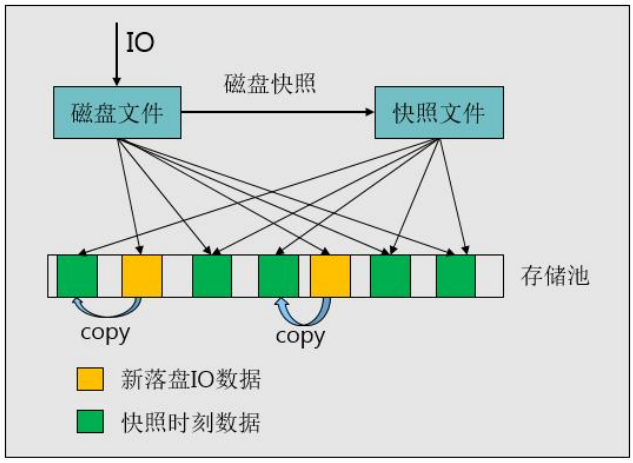


图 4.1.3-3 虚拟磁盘快照（2.0 版本虚拟机）

4.1.3.3 磁盘多副本

磁盘多副本是在磁盘层面实现一次写入多个副本磁盘的操作，通过将 IO 操作同步写入到多个副本磁盘来实现数据的实时备份。用户需要单独配置一个备份存储池，用来存放多副本磁盘的副本，目前支持两副本，即虚拟机的磁盘和一块副本磁盘。当虚拟机磁盘所在的存储或备份存储中的任一存储池出现故障时，可以通过正常的存储池中的磁盘快速恢复虚拟机，保证用户业务的正常运行。

在对虚拟机开启磁盘多副本之后，系统会自动为该虚拟机的所有磁盘在指定的备份存储池上创建一块对应的副本磁盘，并将虚拟机当前磁盘的数据同步到副本磁盘。

开启磁盘多副本的虚拟机开机时，会首先检查虚拟机每块磁盘与其对应副本磁盘的数据一致性，如果不一致的话，提示用户执行同步数据的操作。当用户同步完成磁盘数据，才可以继续开启虚拟机。

开启磁盘多副本的虚拟机开机之后，使用方式与普通虚拟机相同，由于每次数据写入都需要保证在多个磁盘副本中写入成功才算成功，因此，对虚拟机的读写性能会有少许影响。

建议使用性能相同的两个存储池，一个作为虚拟机的工作存储池，一个作为备份存储池。如果两个存储池的性能不一样，建议将性能高的存储池作为虚拟机

的工作存储池，性能低的作为备份存储池。

4.1.3.4 磁盘多队列

磁盘多队列功能可根据队列数成倍增加虚拟机磁盘 I/O 通道，在存储性能充裕的情况下，较传统虚拟机磁盘数据单 I/O 通道，可大大提升虚拟机 I/O 性能，目前可配置 VIRTIO 磁盘接口的虚拟磁盘。

4.1.3.5 磁盘簇大小

虚拟磁盘文件是由多个固定大小的单元组织构成，这些单元被称为 cluster（簇），我们这里可以根据需要，在创建磁盘文件时，自定义 cluster size（簇大小），取值范围为：64K、128K、256K、512K、1M、2M 。cluster size 的大小决定了磁盘 L1、L2 cache 检索的效率，cluster size 越大的话，检索效率越高，磁盘性能越好。目前可配置 qcow2 格式的虚拟磁盘。

4.1.3.6 磁盘二级缓存

虚拟磁盘的二级缓存是使用主机内存，用来缓存虚拟磁盘的部分 L2 表信息，加快虚拟磁盘数据的索引，提升虚拟磁盘的性能。目前可配置 qcow2 格式的虚拟磁盘。

4.1.3.7 磁盘内核 IO 加速

内核 IO 加速功能是虚拟磁盘的一个调优选项，可以针对 virtio 接口的虚拟磁盘进行性能优化。内核 IO 加速将虚拟磁盘的 IO 驱动从用户态迁移到内核态，作为一个独立的内核模块存在，将虚拟磁盘的 IO 驱动与 qemu 解耦合。虚拟磁盘的 IO 路径跳过 qemu 后，就减少了 qemu 和内核之间的上下文切换开销，从而提高 IO 性能。

4.1.3.8 基于虚拟磁盘文件创建虚拟机

InCloud Sphere 提供了基于 qcow2、raw、vmdk 虚拟磁盘创建虚拟机的功能。当前一些 iso 的对外发布通常以镜像文件的方式进行，InCloud Sphere 提供

了应对类似场景的能力。方便用户便捷、快速、有效的方式完成环境的准备以及搭建工作。

4.1.3.9 基于磁盘快照新建虚拟机

InCloud Sphere 支持使用磁盘快照恢复成一个新的虚拟机，基于该功能可以查看虚拟机快照时刻的数据内容，并且不影响原虚拟机的运行。

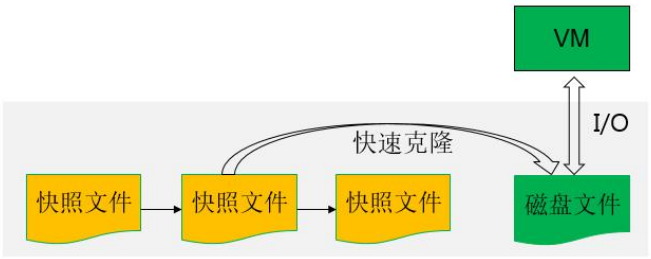


图 4.1.3-4 基于磁盘快照新建虚拟机

4.1.4 虚拟网卡

4.1.4.1 网卡绑定

网卡绑定可以对两个或者更多 NIC 配置在一起，形成一个逻辑网卡，从而提高 InCloud Sphere 主机的恢复能力和传输带宽。

InCloud Sphere 支持的绑定模式有主动-主动、主动-被动和 LACP 模式。

- 主动-主动绑定模式：

主动-主动绑定模式中两个网卡可以同时工作，转发 VM 的业务流量。

如图 4.1.4-1 在管理网络中，NIC2 处于主动状态，NIC 1、3 和 4 处于被动状态。对于 VM 通信，绑定中的所有四个网卡都处于主动状态；在管理或存储通信中绑定只有一个链路 NIC 处于活动状态，其他 NIC 保持未使用状态，除非通信故障转移到其他 NIC。如上图在存储流量中只有 NIC1 处于活动状态，其他 NIC 无流量。

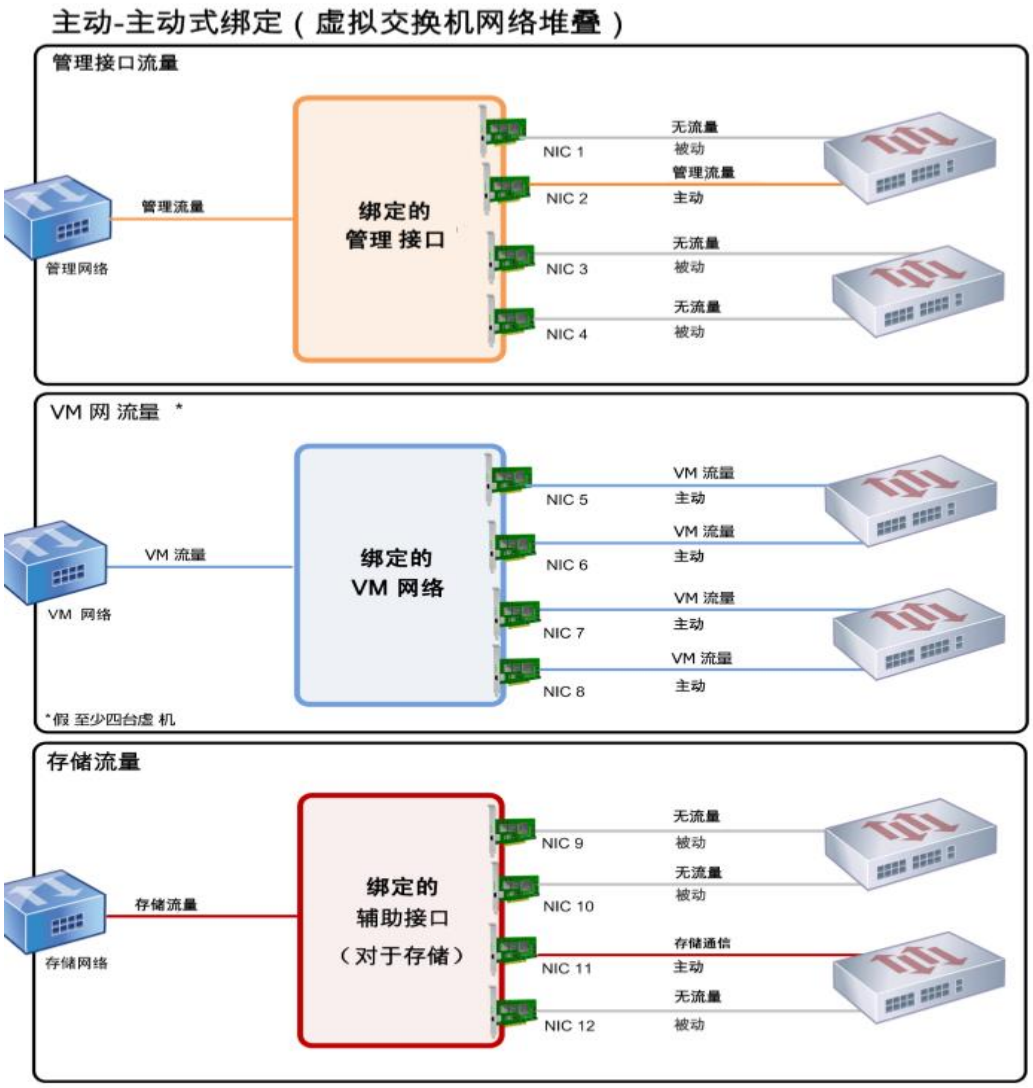


图 4.1.4-1 主动-主动绑定

- 主动-被动绑定

主动-被动绑定仅在一个 NIC 上路由通信，且不会实施负载平衡，即在主动 NIC 上路由通信；当主动 NIC 发生故障时，通信才转移到被动 NIC，网桥上行链路默认的绑定模式是主动-被动绑定。

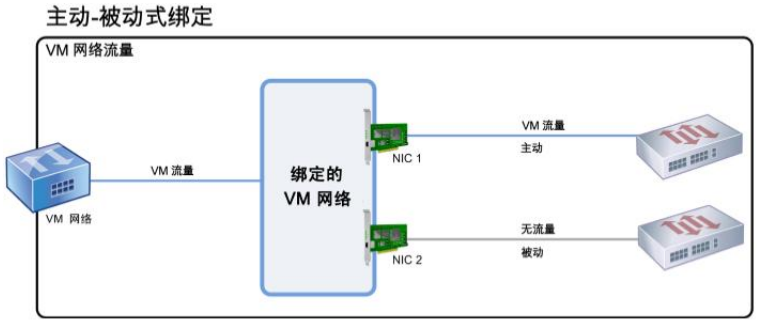


图 4.1.4-2 主动-被动绑定

图 4.1.4-2 中显示了在主动-被动模式下绑定的两个 NIC，NIC1 处于活动状态，此绑定包含一个连接到第二台交换机用于故障转移的 NIC。此 NIC2 仅在 NIC1 发生故障时使用。

主动-被动模式是恢复能力的良好选择，因为它可以提供多种优势。使用主动-被动绑定时，通信不会再在 NIC 之间移动。同样，主动-被动绑定的 NIC 可以分别接入两台交换机以实现冗余，这两台交换机请勿做堆叠，堆叠交换机存在单点故障，破坏了主动-被动绑定的冗余设计。

InCloud Sphere 中主动-被动模式不要求交换机支持 Ether Channel 或 802.3ad（LACP）。建议不需要负载平衡或希望在一个 NIC 上传输通信时，请考虑配置主动-被动模式。

- LACP 链路聚合控制协议绑定

LACP 链路聚合控制协议是一个绑定类型，可将一组端口绑定在一起形成一个逻辑通道使用。LACP 绑定可以提供故障转移，并能增加可用带宽总量。

LACP 绑定有两种类型，分别是：

- 基于源和目的 IP 和端口的负载均衡
- 基于源 MAC 地址和 VLAN 的负载平衡

基于源和目的地址 IP 和端口的负载均衡：LACP 默认绑定散列算法，在虚拟机运行多个应用程序，且每个应用程序使用不同的 IP 或端口号，此散列类型可以在若干链路上分布通信，使来宾系统有可能使用聚合吞吐量。

如图 4.1.4-3，散列类型能够使虚拟机上两个不同应用程序的通信分布到两个不同的 NIC 上。

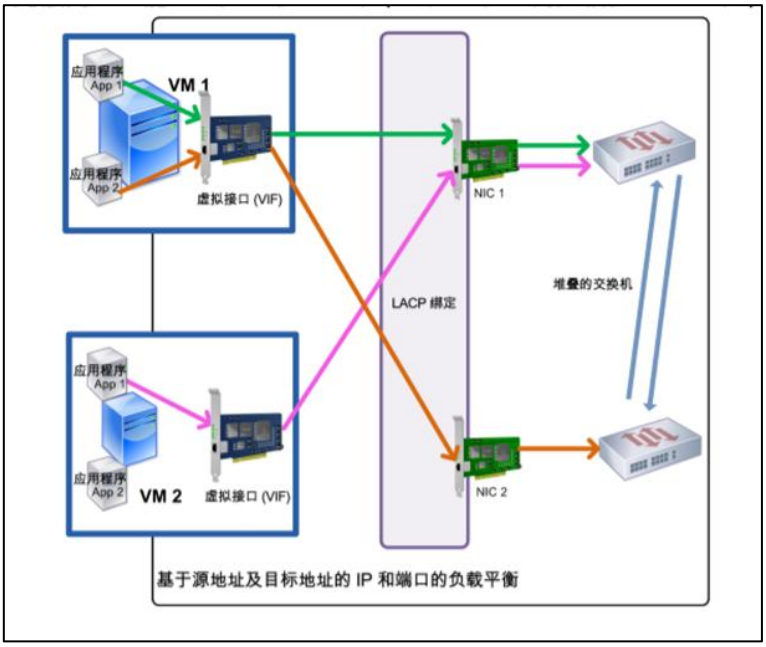


图 4.1.4-3 基于源和目的地址 IP 和端口的负载均衡

基于源 MAC 地址和 VLAN 的负载平衡：当同一个主机上有多个虚拟机时，此类型的负载平衡可以发挥很好的作用。通信平衡根据发起通信的 VM 的虚拟 MAC 地址实施。

图 4.1.4-4 显示如果使用 LACP 绑定并启用将基于源 MAC 地址作为散列类型的 LACP，如果 NIC 的数量超过虚拟机网卡的数量，则不会使用所有的 NIC。因为有三个 NIC 和两个 VM，所以只有两个 NIC 能同时使用，因而无法达到最大绑定吞吐量。

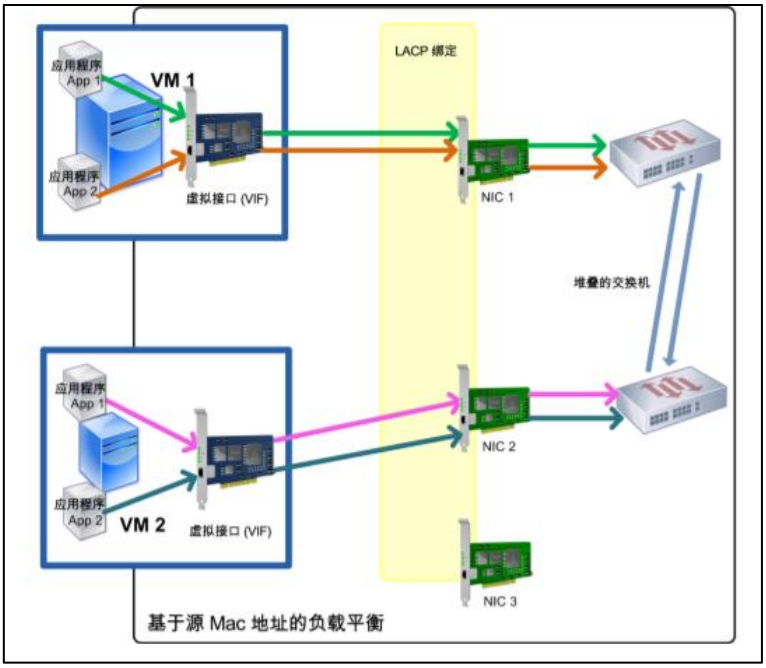


图 4.1.4-4 基于源 MAC 地址和 VLAN 的负载平衡

4.1.4.2 虚拟网卡 QoS

要限制 VM 每秒可以发送的传输数据量，可以针对 VM 虚拟网卡设置可选的服务质量 (QoS) 值。该设置允许传出的数据包指定最大传输速率（以每秒千比特为单位）。 QoS 值将限制 VM 的上行和下行传输速率。

4.1.4.3 SR-IOV

InCloud Sphere 支持虚拟机挂载 SR-IOV 网卡，SR-IOV 技术是一种基于硬件的虚拟化解决方案,SR-IOV 实现了多个虚拟机共享使用一个物理网卡的功能，并且达到直接分配的目的，提供了网络交换的性能。

SR-IOV 使用两种功能（Function）：物理功能（Physical Functions，PF），PF 是完整的带有 SR-IOV 能力的 PCIe 设备,PF 能像普通 PCI 设备那样被发现、管理和配置;虚拟功能（Virtual Functions，VF）：简单的 PCIe 功能，VF 只能处理 I/O,每个 VF 都是从 PF 中分离出来的。每个物理硬件都有一定数目的 VF 可以配置，一个 PF 能被虚拟成多个 VF 用于分配给多个虚拟机，Hypervisor 能将一个或者多个 VF 分配给一个虚拟机。在某一时刻，一个 VF

只能被分配给一个虚拟机，一个虚拟机可以拥有多个 VF。在虚拟机的操作系统看来，一个 VF 网卡看起来和一个普通网卡没有区别。

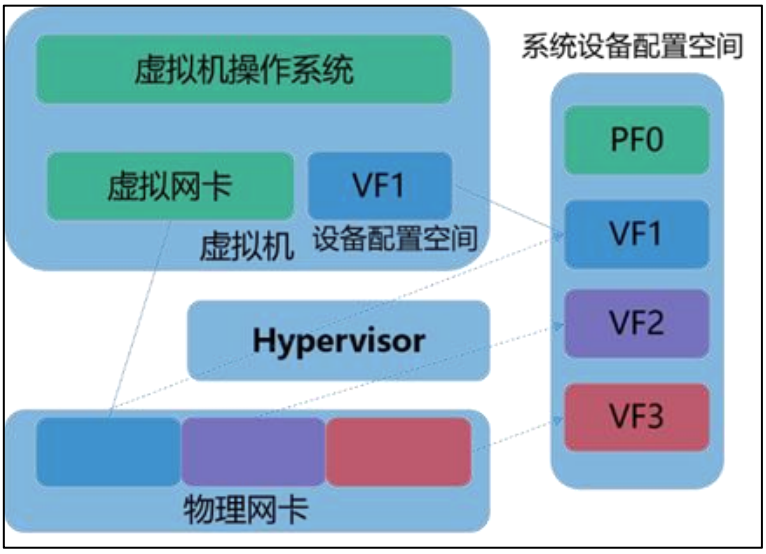


图 4.1.4-5 SR-IOV 技术原理图

SR-IOV 特性的使用需要硬件平台 Intel VT-d 或者 AMD IOMMU 的支持,在使用此功能时,这些特性必须在 BIOS 中被启用,如果 BIOS 中有 SR-IOV 的使能配置，那么也需要进行开启操作，系统支持虚拟机在使用 SR-IOV 类型的网卡时，进行热添加，不支持 SR-IOV 类型网卡的热删除以及虚拟机使用 SR-IOV 类型网卡时的迁移功能。

4.1.4.4 MACVTAP

InCloud Sphere 支持虚拟机挂载 MACVTAP 虚拟网卡，MACVTAP 是 Linux 引入的一种新的网络设备模型，用以代替传统的 Linux TAP 设备加 Bridge 设备组合，同时支持新的虚拟化网络技术。MACVTAP 可以直接绕过内核协议栈，有效提高数据从物理网卡到虚拟机之间的性能，并简化虚拟化环境中的交换网络。MACVTAP 的实现基于传统的 MACVLAN，和 TAP 设备一样，每一个 MACVTAP 设备拥有一个对应的 Linux 字符设备，并拥有和 TAP 设备一样的 IOCTL 接口，因此能直接被 KVM/Qemu 使用，方便地完成网络数

据交换工作。

4.1.4.5 网卡 Pass-through

网卡 Pass-through 也称为网卡直通，通过 Pass-through，虚拟机可以使用 I/O 内存管理单元直接对物理网卡进行独占使用，简单理解就是在虚拟机添加网卡时将物理网卡以透传的方式直接给虚拟机使用，即支持客户机以独占方式访问宿主机上的物理网卡，客户机对该物理网卡的 I/O 交互操作和实际的物理设备操作完全一样，不需要或者很少需要 KVM 的参与。

网卡的 Pass-through 需要硬件平台 Intel VT-d 或者 AMD IOMMU 的支持，在使用此功能时，这些特性必须在 BIOS 中被启用，系统支持虚拟机在使用网卡直通时，进行热添加，不支持 Pass-through 类型网卡的热删除以及虚拟机使用 Pass-through 类型网卡时的迁移功能。

4.1.4.6 IP/MAC 绑定

IP/MAC 地址绑定应用在虚拟网卡上，可以有效防止虚拟机用户私自修改 IP 或 MAC 来发起 IP、MAC 仿冒攻击。启用了 IP/MAC 绑定后，只有匹配了 IP 和 MAC 的网络流量才能进行正常网络通信，如果虚拟机内部更改了 IP 地址或者 MAC 地址，网络流量中断无法进行正常业务通信。

4.1.4.7 虚拟网卡多队列

虚拟机使用 virtio 类型网卡时，支持配置虚拟机网卡多队列，默认的 virtio-net 不能并行地传送或者接收网络包，因为默认 virtio_net 只有一个 TX 和 RX 队列。而多队列 virtio-net 提供了一个随着虚拟机的虚拟 CPU 增加而增强网络性能的方法，使得 virtio 可以同时使用多个 virt-queue 队列，将网卡中断分散给不同的 CPU 处理，提高虚拟机网络的吞吐量。

虚拟网卡多队列功能在以下情况下具有明显优势：

1) 网络流量非常大

2) 虚拟机同时有非常多的网络连接，包括虚拟机之间的、虚拟机到主机的、虚拟机到外部系统的等

3) virtio 队列的数目和虚拟机的虚拟 CPU 数目相同，这是因为多队列能够使得一个队列独占一个虚拟 CPU。

4.1.5 GPU 虚拟化

4.1.5.1 GPU 直通

GPU 设备属于主机 pci 设备，其直通原理同其他类型的 pci 设备直通，如网卡直通。将物理 GPU 设备与主机驱动分离后，与 vfio-pci 驱动绑定。虚拟机添加对应的 pci 编号的主机 pci 设备，最终完成 pci 设备的直通到虚拟机。

物理设备直通后需要在虚拟机中安装相应的 GPU 驱动设备才能正常工作。同时虚拟机 CPU 模式应设置为 host-model，即接近主机 CPU 特性的模式。

GPU 直通技术让用户能够创建一个虚拟工作站，即可享受到专用图形处理器的所有优势。虚拟工作站借助 GPU 加速技术提供工作站级别的用户体验，用于高端图形设计、应用仿真、3D 图形渲染、数据分析等工作负载加速。

4.1.5.2 vGPU

NVIDIA GPU 虚拟化技术允许多台虚拟机同时使用一块 GPU 设备的计算资源，提供卓越的图形、计算性能以及良好的软件兼容性。

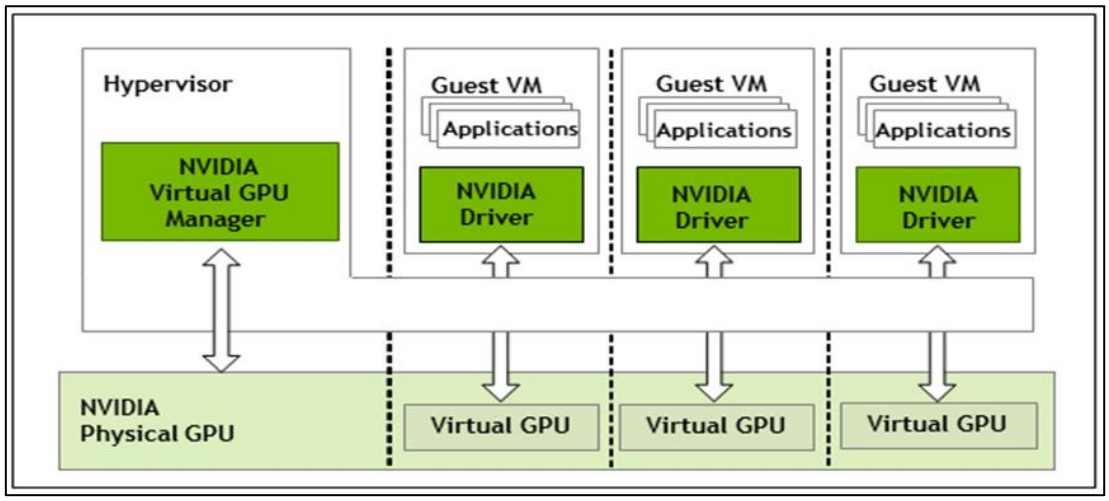


图 4.1.5-1 vGPU 原理

InCloud Sphere 基于 GPU 组的概念统一管理集群中的 GPU 设备，基于集群创建一个或多个 GPU 组，每个 GPU 组都有对应的 GPU 设备型号以及 GPU 设备使用方式，使用方式包括直通和 vGPU 两种，选择虚拟化的使用方式，并指定 vGPU 类型，每一类 vGPU 都对应了 GPU 的不同功能、API 的适用范围以及可生成 vGPU 的数量，用户可根据自身应用需求来选择某一类 vGPU 类型。

创建 GPU 组之后，可以添加 GPU 组对应型号的集群 GPU 设备，此时需要集群中提供 GPU 设备的主机安装 NVIDIA vGPU Manager 驱动。GPU 组添加 GPU 设备之后，会自动根据 GPU 组属性中的 vGPU 类型自动生成 vGPU 设备。

GPU 组添加 GPU 设备之后，虚拟机可在关机状态下添加 GPU 设备，选择相应的 GPU 组并点击完成即可成功添加 vGPU 设备。

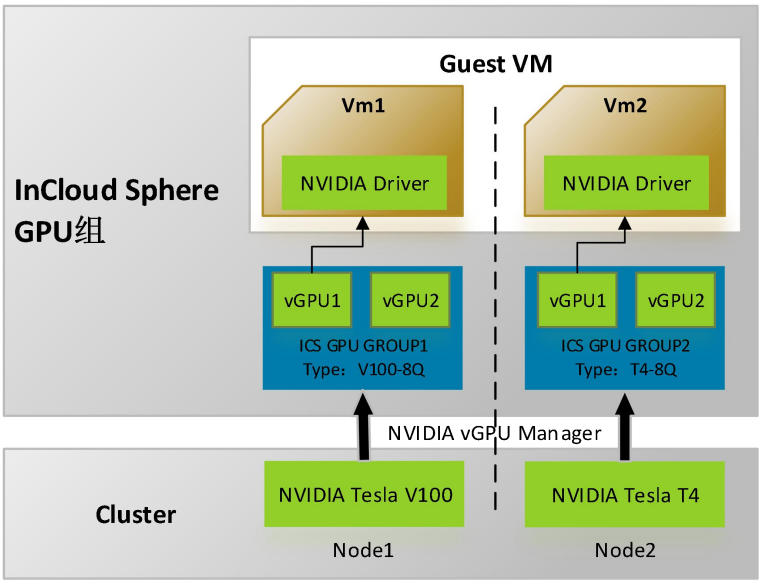


图 4.1.5-2 GPU 组管理

4.1.6 USB 设备

4.1.6.1 USB 设备直通

InCloud Sphere USB 直通是指将 USB 设备直接分配给虚拟机，此后，虚拟

机中的应用程序可直接访问 USB 设备，而不需要通过 VMM 进行管理。虚拟机启动前，把设备从 VMM 中注销，然后启动虚拟机，在虚拟机的设备驱动中注册该设备。InCloud Sphere 支持 USB 设备的管理与直通并支持选择按 USB 协议或者自适应挂载 USB 设备，同时挂载 USB 设备的虚拟机支持 HA 高可用。

4.1.6.2 USB 网络映射

InCloud Sphere 系统中所有主机均运行 USB-redir server 服务监听对应的网络端口号，虚拟机添加 redirdev USB 设备作为 client 端，client 端监听对应远程主机的端口号。这样在 server 与 client 之间建立网络连接，监听并处理 USB 请求或者数据，最终网络 USB 映射方案。

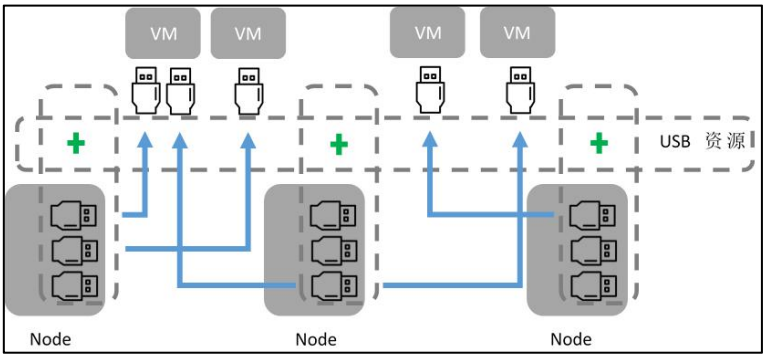


图 4.1.6-1 USB 资源池管理

InCloud Sphere USB 资源池管理支持 USB 与网络 USB 相互转换，前置条件是 USB 均处于空闲状态。

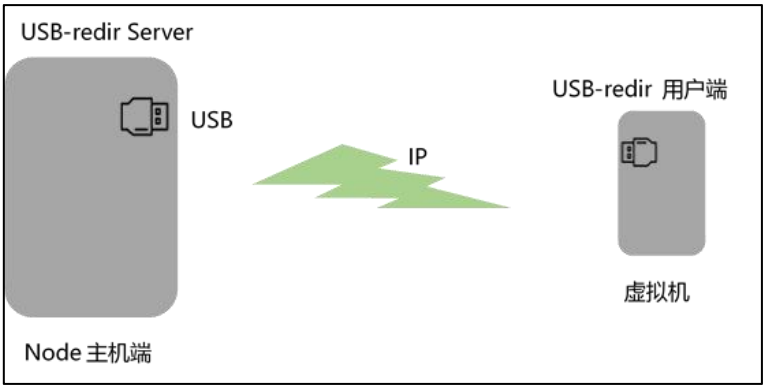


图 4.1.6-2 USB over IP

4.1.7 软驱设备

InCloud Sphere 支持为虚拟机挂载虚拟软驱设备，来将软驱设备透传给虚拟机。用于安装虚拟机的操作系统、业务软件或者驱动程序。

4.1.8 虚拟显卡

虚拟机显卡类型可分为 Qxl、Vga、Cirrus 三种类型，分别可支持 16MB-512MB 显存大小，通过调节显卡类型及显存，根据相应虚拟机操作系统类型，可支持 4K 以上分辨率。

4.1.9 看门狗

虚拟机支持增加看门狗设备来保证虚拟机系统和应用的高可用性。增加看门狗设备后，系统会监控看门狗计时器，当计时器超时后，系统会对虚拟机执行重启操作。当前系统支持的看门狗类型为 Intel 6300ESB。

4.1.10 PCI 设备自定义配置

InCloud Sphere 支持直通主机所有可用 PCI 设备。该功能通过自动识别所有 PCI-Bridge 插槽上的设备，将主机端所有可用 PCI 设备都整合起来，统一在 iCenter 的计算池中进行管理。

在所有 PCI 设备中，有一类特殊的 multi-function 设备，该类设备表现为多个 PCI 设备集成于同一物理 PCI 设备中，且 PCI 设备之间可以共享内存数据，因此为保证虚拟机数据的隔离性，InCloud Sphere 会将这些 PCI 设备归纳到同一 iommu_group 组里面，虚拟机直通这些设备时，需要同时将这一 iommu_group 组中的设备全部直通到同一台虚拟机以防止虚拟机内部通过 multi-function 设备的特性访问主机或其他虚拟机中的数据。

4.1.11 UEFI 启动

InCloud Sphere 支持以 UEFI 的形式引导启动虚拟机，UEFI，全称 Unified

Extensible Firmware Interface，即“统一的可扩展固件接口”，是一种详细描述全新类型接口的标准，UEFI 抛去了传统 BIOS 需要长时间自检的问题，让硬件初始化以及引导系统变得简洁快速。UEFI 缩短了启动时间和从休眠状态恢复的时间，支持容量超过 2 TB 的驱动器，支持 64 位的现代固件设备驱动程序，系统在启动过程中可以使用它们来对超过 172 亿 GB 的内存进行寻址。

4.1.12 虚拟机回收站

用户可以选择将待删除的虚拟机放入回收站，以备用户在必要时恢复该虚拟机。可以设置虚拟机在回收站中的保留时长，超过该时长，则系统会彻底销毁该虚拟机。

进入回收站的虚拟机依然保留分配给它的存储资源和透传的设备，只有虚拟机被销毁后，其占用的硬件资源才会被释放。

4.1.13 虚拟机串口

虚拟机可以增加串口设备，可以将虚拟机的串口 COM1（Linux 系统为 ttyS0）映射为所在主机的某一个指定的端口，客户端可以通过 Telnet 连接主机端口来连接虚拟机的 COM1(Linux 系统为 ttyS0)串口，此种方式为 Telnet 模式。虚拟机的串口也支持服务端和客户端的模式，将两个虚拟机的串口 COM2（Linux 系统为 ttyS1）连接起来，其中一个虚拟机作为服务端，另一个虚拟机作为客户端，客户端服务端的模式用于两个在同一个系统上的两个不同的虚拟机之间相互连接。

4.1.14 一云多芯

一云多芯即在同一个 iCenter 下可以同时纳管不同架构的计算节点，包括 X86（Intel、AMD），C86（海光），ARM（鲲鹏、飞腾）等，以及这些不同架构的虚拟机。

iCenter 通过对不同主机指令集的梳理，虚拟机可以在兼容的主机间进行离

线或在线的计算迁移。

4.2 安全容器

4.2.1 基础技术原理

4.2.1.1 容器（Container）

容器是一种沙盒技术，主要目的是为了将应用运行在其中，与外界隔离；及方便这个沙盒可以被转移到其它宿主机器。本质上，它是一个特殊的进程。

通俗点的理解就是一个装应用软件的箱子，箱子里面有软件运行所需的依赖库和配置。开发人员可以把这个箱子搬到任何机器上，且不影响里面软件的运行。

容器技术是一种轻量级的操作系统虚拟化技术（虚拟化技术通过 Hypervisor 实现 VM 与底层硬件的解耦），将应用程序及其运行依赖环境打包封装到标准化、强移植的镜像中，通过容器引擎提供进程隔离、资源可限制的运行环境，实现应用与 OS 平台及底层硬件的解耦，一次打包，随处运行。容器基于镜像运行，可部署在物理机或虚拟机上，通过容器引擎与容器编排调度平台实现容器化应用的生命周期管理。

4.2.1.2 安全容器（Kata Containers）

虚拟机通常包括整个操作系统和应用程序，里面运行的是一个真实的操作系统。本质上虚拟机是 Hypervisor 虚拟化出来的硬件上安装不同的操作系统，而容器是宿主机上运行的不同进程。从用户体验上来看，虚拟机是重量级的，占用物理资源多，启动时间长。容器则占用物理资源少，启动迅速。相对地，虚拟机隔离的更彻底，容器则要差一些。

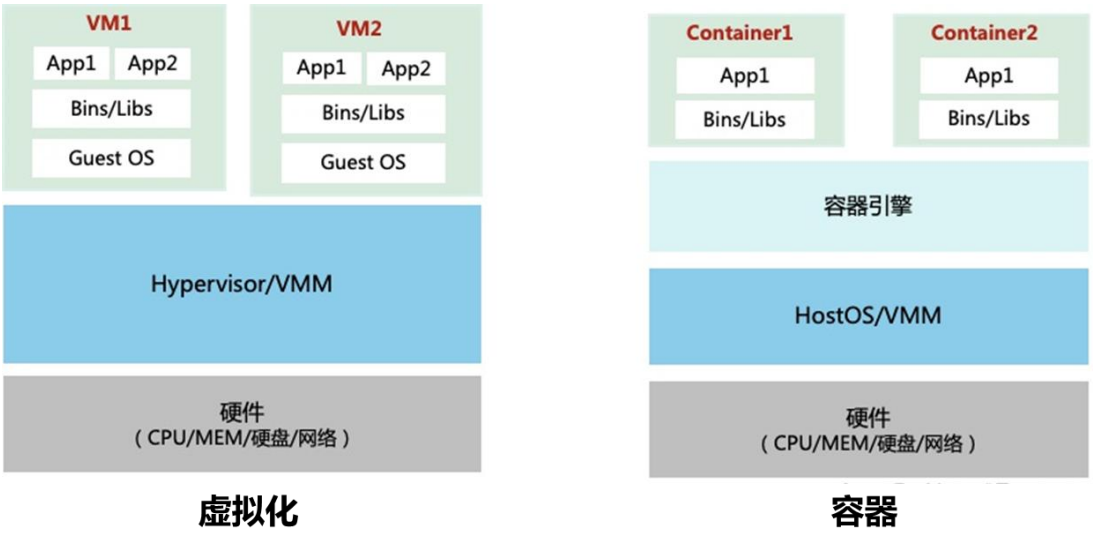


图 4.2.1.2-1 虚拟机与容器技术架构

VM 中包含 GuestOS，调度与资源占用都比较重。而容器仅包含应用运行所需的文件，管理容器就是管理应用本身。如表 1，容器具有极其轻量、秒级部署、易于移植、敏捷弹性伸缩等优势；但 VM 是 OS 系统级隔离，而容器是进程级隔离，容器的安全性相对更弱一些，需要一些额外的安全技术或安全容器方案来弥补。作为云原生的核心技术，容器、微服务与 DevOps/ CICD）等技术已成为应用架构转型或实现技术中台不可或缺组件。

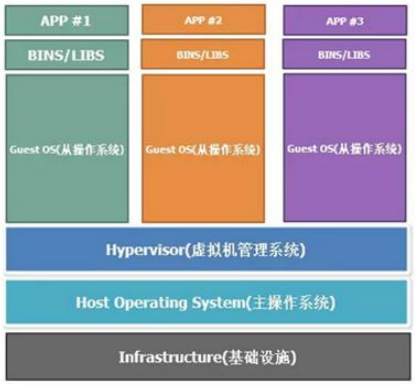
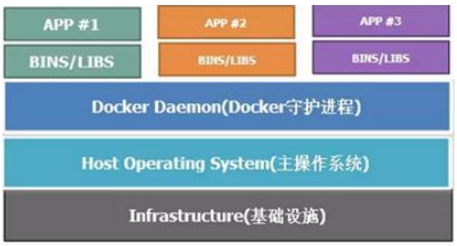
架构	虚拟机 	容器 
启动	分钟级	秒级
资源占用量	大	小（没有臃肿的从操作系统）
内存占用	一般为GB	一般为MB
运行密度	几个	单机支持上百个容器
拥有独立操作系统	是	否，共用宿主机内核
隔离性	更强（虚拟硬件级全隔离）	文件系统、进程及网络隔离
监控粒度	资源级	应用级

图 4.2.1.2-2 虚拟机与容器技术比较

由于传统 runc 容器技术是共享宿主机内核的，所以不可避免带来容器逃逸等安全性问题，以 Kata Containers 为代表的虚拟机级别的安全隔离技术，逐渐成为了安全容器技术的主流。

基于 Kata Containers 的安全容器技术，每个 Container/POD 都被单独的虚拟机隔离，遵循 OCI 开放容器接口，虚拟机对用户不可见，采用轻量级虚拟机，兼具容器的速度和虚拟机的隔离性，另与 Docker 及 Kubernetes 无缝集成，用户的容器应用无需任何改动。

4.2.1.3 容器镜像（Image）

容器镜像是一个只读的模板，含有启动容器所需要的文件系统及其内容。比如一个镜像可以包含一个运行在 Apache 上的 Web 应用和其使用的 Ubuntu 操作

系统。

4.2.1.4 容器镜像库（Image Registry）

容器镜像库是用来存储容器镜像的存储池，具备存储、分发等管理镜像的能力。

镜像仓库分为公共镜像仓库和私有镜像仓库。Docker Hub 是 Docker 官方的公开镜像仓库，有很多应用或者操作系统的官方镜像，还有很多组织或者个人开发的镜像免费存放、下载、研究和使用。除了公开镜像仓库，也可以使用 docker-registry 或微软开源的 Harbor 来构建自己的私有镜像仓库。

4.2.1.5 Kubernetes

Kubernetes 是一个可移植、可扩展的开源平台，用于管理容器化的工作负载和服务，可促进声明式配置和自动化。 Kubernetes 拥有一个庞大且快速增长的生态，其服务、支持和工具的使用范围相当广泛。

容器是打包和运行应用程序的好方式。在生产环境中， 我们需要管理运行着应用程序的容器，并确保服务不会下线。 例如，如果一个容器发生故障，则你需要启动另一个容器。 Kubernetes 为我们提供了一个可弹性运行分布式系统的框架。Kubernetes 会满足你的扩展要求、故障转移你的应用、提供部署模式等。

Kubernetes 可以提供：

服务发现和负载均衡，Kubernetes 可以使用 DNS 名称或自己的 IP 地址来曝露容器。 如果进入容器的流量很大， Kubernetes 可以负载均衡并分配网络流量，从而使部署稳定。

存储编排，Kubernetes 允许你自动挂载你选择的存储系统，例如本地存储、公共云提供商等。

自动部署和回滚，你可以使用 Kubernetes 描述已部署容器的所需状态，它

可以以受控的速率将实际状态更改为期望状态。 例如，你可以自动化 Kubernetes 来为你的部署创建新容器，删除现有容器并将它们的所有资源用于新容器。

自动完成装箱计算，你为 Kubernetes 提供许多节点组成的集群，在这个集群上运行容器化的任务。 你告诉 Kubernetes 每个容器需要多少 CPU 和内存 (RAM)。 Kubernetes 可以将这些容器按实际情况调度到你的节点上，以最佳方式利用你的资源。

自我修复，Kubernetes 将重新启动失败的容器、替换容器、杀死不响应用户定义的运行状况检查的容器， 并且在准备好服务之前不将其通告给客户端。

密钥与配置管理，Kubernetes 允许你存储和管理敏感信息，例如密码、OAuth 令牌和 ssh 密钥。 你可以在不重建容器镜像的情况下部署和更新密钥和应用程序配置，也无需在堆栈配置中暴露密钥。

4.2.1.6 K3s

K3s 是 Rancher 开源的、经过 CNCF 云原生基金会一致性认证的、**轻量**的生产级 Kubernetes 发行版。目前主要应用场景：边缘（Edge）、物联网（IoT）、持续集成/交付（CI/CD）、ARM 架构。

K3s 当前社区比较活跃，有 1700 多个贡献者，版本发布紧跟 Kubernetes 社区版本；内置 storage provider、ingress 等插件；支持单节点或高可用部署，支持第三方数据库，另支持多 CPU 架构。

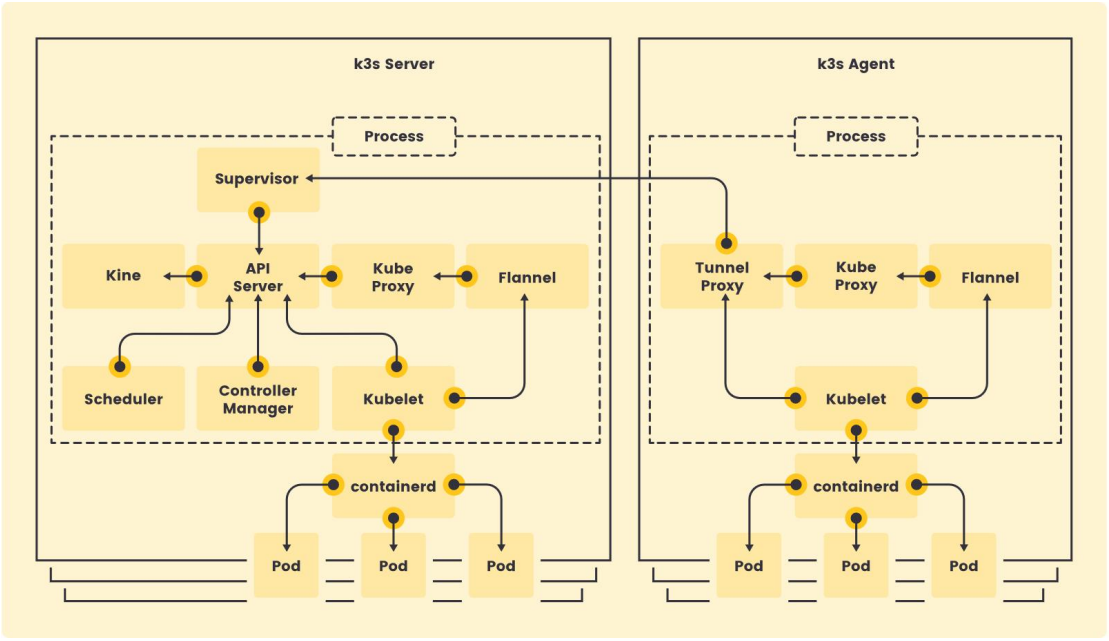


图 4.2.1.6 K3s 技术架构

4.2.1.7 容器实例（Pod）

容器实例对应于 Kubernetes 中的 Pod 概念，Pod 是可以在 Kubernetes 中创建和管理的、最小的可部署的计算单元。Pod 通常包含一个或多个容器，其中的容器将被调度到同一台宿主机中，共享计算、网络和存储资源。

4.2.1.8 容器存储插件（CSI）

容器存储接口已成为容器持久卷管理的行业标准。它在 2017 春季开始的，在云原生计算基金会（CNCF）的赞助下，作为一个由来自领先的容器编排系统 Kubernetes、MeSOS、DOCKER 和 Cloud Foundry 的社区成员的协同合作。该规范的最终 v1.0 版本发布于 2018 年。

CSI 规范了存储管理，包括容量提供、监视和将卷连接到主机。CSI 是一种非常灵活的机制，能够支持各种底层存储系统实现，包括 DAS、NAS、SAN 和分立的存储。此外，CSI 插件对它们提供给其他容器的持久卷的性能没有负面影响。

SC（Storage Class）是存储资源的抽象定义，对用户设置的 PVC 申请屏蔽

后端存储的细节，一方面减少了用户对于存储资源细节的关注，另一方面减轻了管理员手工管理 PV 的压力，转由系统自动完成 PV 的创建和绑定，实现动态的资源供应。SC 资源对象的定义主要包括名称、后端存储的提供者及相关参数配置和回收策略。

持久卷（PersistentVolume，PV）是集群中的一块存储，可以由管理员事先制备，或者使用存储类（Storage Class）来动态制备。持久卷是集群资源，就像节点也是集群资源一样。PV 持久卷和普通的 Volume 一样，也是使用卷插件来实现的，只是它们拥有独立于任何使用 PV 的 Pod 的生命周期。

持久卷声明（PersistentVolumeClaim，PVC）表达的是用户对存储的请求。概念上与 Pod 类似。Pod 会耗用节点资源，而 PVC 声明会耗用 PV 资源。Pod 可以请求特定数量的资源（CPU 和内存）；同样 PVC 声明也可以请求特定的大小和访问模式（例如，可以要求 PV 卷能够以只读或读写模式之一来挂载）。

4.2.1.9 容器网络插件（CNI）

CNI（container network interface）的目的在于定义一个标准的接口规范，使得 Kubernetes 在增删 POD 的时候，能够按照规范向 CNI 实例提供标准的输入并获取标准的输出，再将输出作为 Kubernetes 管理这个 POD 的网络的参考。在满足这个输入输出以及调用标准的 CNI 规范下，Kubernetes 委托 CNI 实例去管理 POD 的网络资源并为 POD 建立互通能力。

CNI 本身实现了一些基本的插件，比如 bridge、ipvlan、macvlan、loopback、vlan 等网络接口管理插件，还有 dhcp、host-local 等 IP 管理插件，并且主流的 container 网络解决方案都有对应 CNI 的支持能力，比如 Calico、Canal、Flannel、Kube-Router、Weave 等。

4.2.2 关键技术点

4.2.2.1 容器镜像库

容器镜像库提供镜像库的初始化等管理服务，以及镜像的上传、拉取等功能。系统内置的容器镜像库由 docker-registry 服务构建，多个 docker-registry 服务由 Kubernetes Service 提供固定浮动 IP（VIP）。

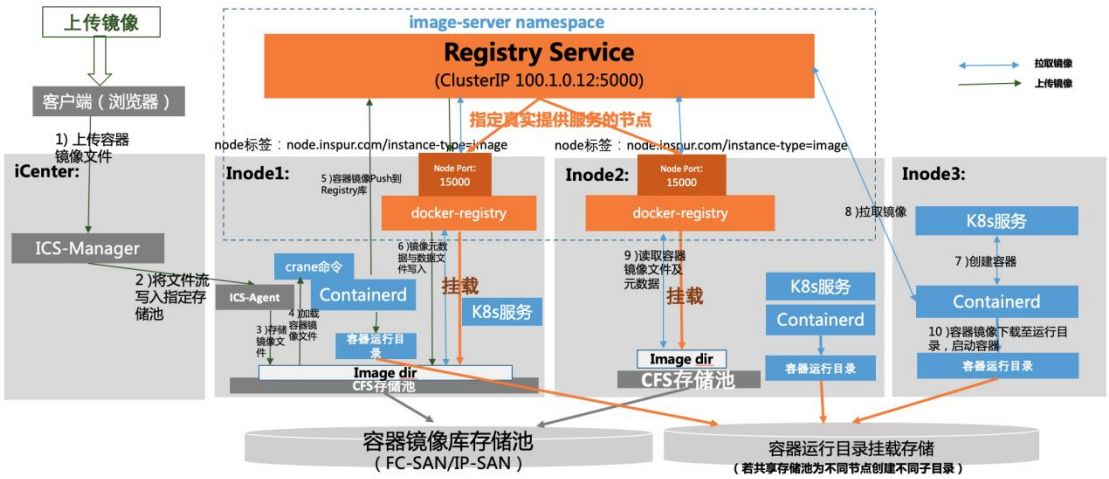


图 4.2.2.1 容器镜像库技术架构

上传镜像流程：

1）用户通过浏览器上传 tar 格式容器镜像

2）ICS-Manager（管理节点服务）将镜像文件流传递至 ICS-Agent（计算节点代理）处理

3）ICS-Agent 将镜像文件流上传至容器镜像库存储池

4）在本地加载镜像文件 push 到 docker-registry 中

5）docker-registry 将容器镜像元数据写入数据库

容器实例使用镜像流程：

1）K3s 服务下发创建容器实例请求

2）Containerd 接收到请求后，将容器镜像从 docker-registry 中下载到本地

容器运行目录

3）容器实例从本地容器运行目录启动

4.2.2.2 容器网络

容器网络的一个基本要求是将容器与 SDN 结合，能够自由灵活地使用基于 Neutron 的 SDN 网络。Kuryr 作为 OpenStack 的一个开源项目，目的是将容器网络与 Neutron 对接。Kuryr 将 Neutron 作为南向接口来与容器网络对接，即 Kuryr 的北向是容器网络接口，南向是 Neutron。

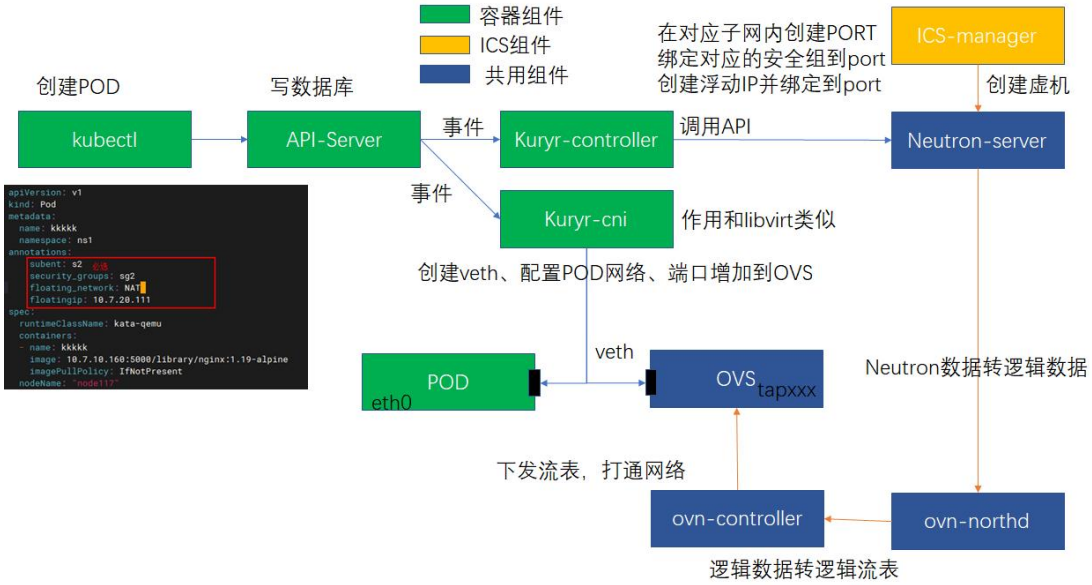


图 4.2.2.2-1 容器网络控制面与数据面

控制面：

VM：ICS-Manager 调用 Neutron API，libvirt 负责创建网卡并插入 OVS。

POD：kuryr-controller 调用 Neutron API，kuryr-cni 负责创建网卡并插入 OVS。

数据面：

由 OVN 下发流表到 OVS，OVS 负责转发。

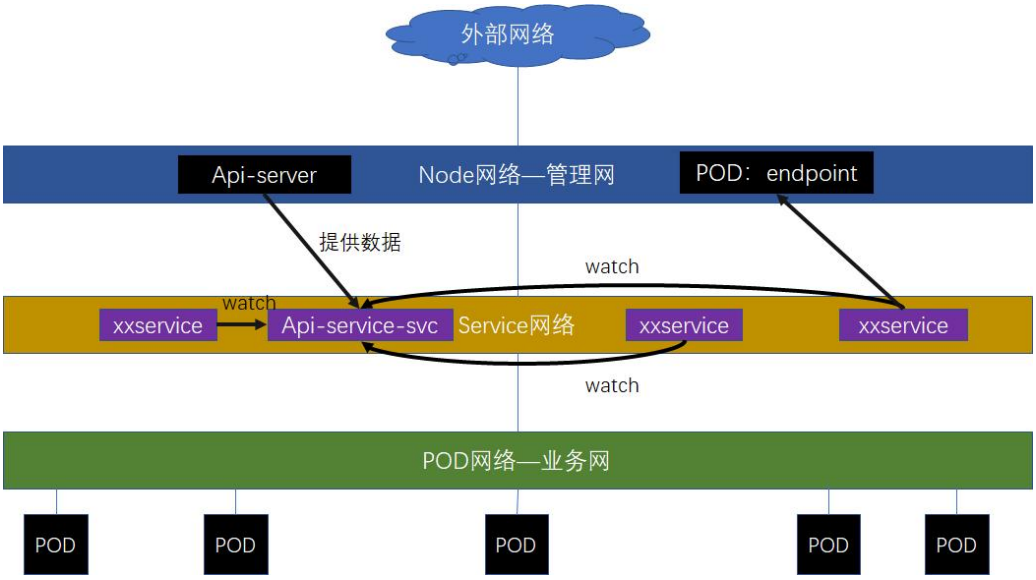


图 4.2.2.2-2 容器三层网络（管理网、Service 网络、POD 网络）

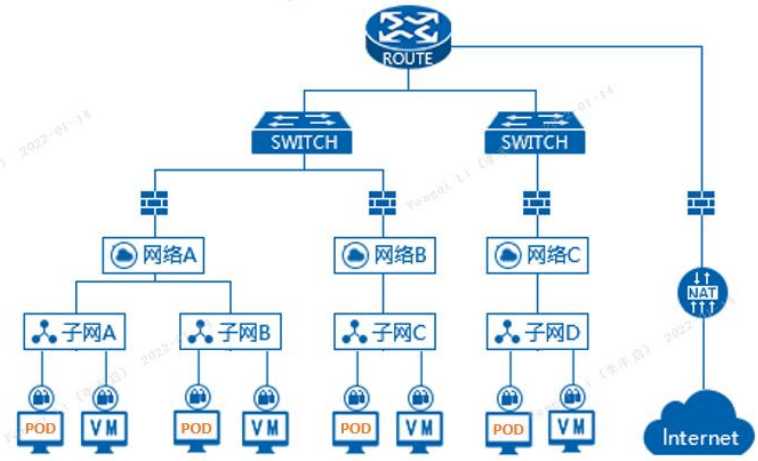


图 4.2.2.2-3 虚拟机与容器网络通信模型

虚拟机与容器网络通信的几种实现方式：

1、虚拟机和容器使用同一个 SDN 网络子网，直接通信

2、虚拟机和容器使用不同的 SDN 网络子网，加路由通信

3、虚拟机使用基础网络，容器使用 SDN 网络，通过容器绑定浮动 IP 通信

4、虚拟机使用基础网络，容器使用 SDN 网络，保证基础网络和 SDN 网络

的 VLAN 一致，CIDR 一致即可通信。

4.2.2.3 容器存储

与容器服务相关的，共有三种存储池：镜像库存储池、容器运行目录存储池和容器卷存储池。

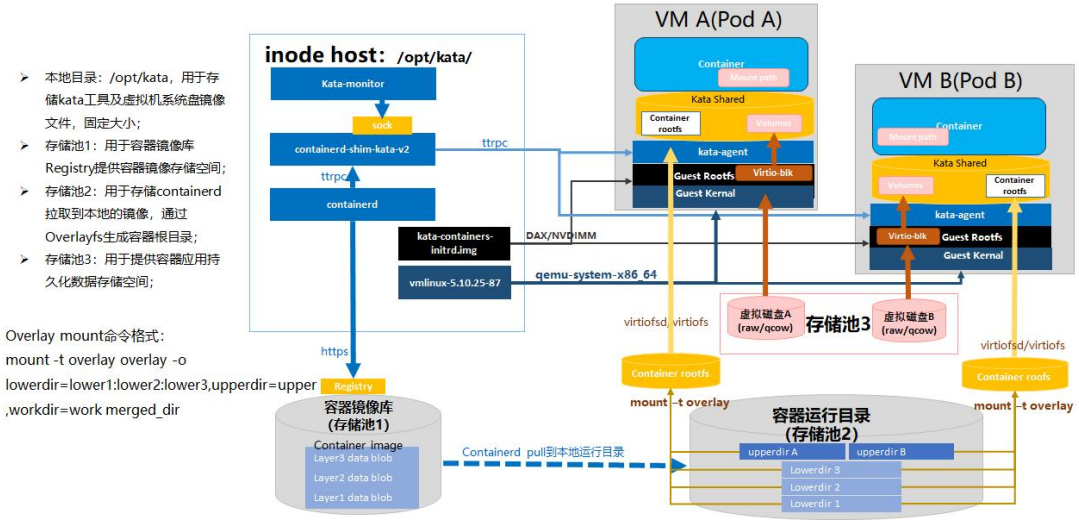


图 4.2.2.3-1 三种容器相关存储池及其原理

容器运行目录存储池，在普通存储池的基础上，创建容器运行目录，此目录作为容器本地镜像的缓存目录，不与其他类型存储池共用。

镜像库存储池，在某个存储池关联的主机运行 **docker-registry**，以提供镜像管理服务，支持镜像的上传、拉取等功能；

容器卷存储池，容器卷存储池与虚拟磁盘存储池共用存储空间，在同一个存储上，既可以创建容器实例用的容器卷，也能创建虚拟机用的虚拟磁盘，原理如下图：

容器存储插件原理：

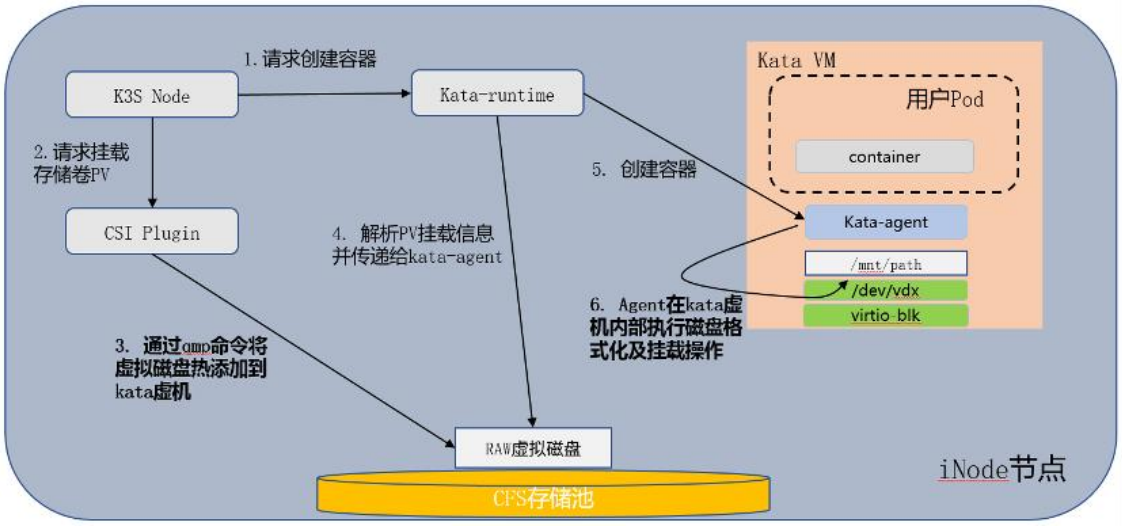


图 4.2.2.3-2 基于虚拟磁盘的容器存储插件原理

基于虚拟磁盘的容器存储插件，由 CSI-Node 组件将 CFS 存储池上的虚拟磁盘添加到 Kata 虚拟机，然后由虚机内部的 Kata-Agent 组件完成对磁盘 vdx 的格式化、挂载等操作。不再依赖于 virtio-fs，有利于简化对存储卷 PV 的管理，提升容器的读写性能。

1）CSI-Controller

调用平台接口完成虚拟磁盘在存储池上的创建、删除、扩容等操作。

2）CSI-Node

在 CSI 插件中通过 qmp blockdev-add/del 等命令对虚拟磁盘进行操作。CSI-Node 完成对 Kata 虚拟机的磁盘添加后，调用 Kata-Runtime 管理命令，通知 Kata-Agent 进行磁盘的格式化、挂载操作。

3）Kata-Runtime

实现对文件系统的管理接口，实现文件系统的格式化、挂载、卸载、扩容、状态查询等功能。

4）Kata-Agent

根据 Kata-Runtime 发起的 grpc 请求，执行具体的文件系统格式化、挂载等操作。

4.2.3 容器实例管理

容器实例对应 Kubernetes 中的 Pod 对象，Pod 是 Kubernetes 创建或部署的最小单位。一个 Pod 通常封装一个容器（container）、存储资源（volume）、一个独立的网络 IP 以及管理控制容器运行方式的策略选项。

超融合平台提供基于 Kata Containers 的安全沙箱容器，一个 Kata 虚拟机可以认为是一个 Pod，可提供虚拟机级别的安全和资源隔离能力，同时针对容器运行环境进行深度优化，提供比虚拟机更快的启动速度和运行效率。

4.2.3.1 启动命令

启动容器就是启动主进程，但有些时候，启动主进程前，需要一些准备工作。比如 MySQL 类的数据库，可能需要一些数据库配置、初始化的工作，这些工作要在最终的 MySQL 服务器运行之前解决。这些操作，可以在制作镜像时通过在 Dockerfile 文件中设置 ENTRYPOINT 或 CMD 来完成，如下所示的 Dockerfile 中设置了 ENTRYPOINT ["ifup", "eth0"]命令，其将会在容器启动时执行。例如：

```
FROM ubuntu
ENTRYPOINT["ifup", "eth0"]
```

在容器实例中也可以设置容器的启动命令来覆盖 Dockerfile 中设置的启动命令，如下图：

界面上设置的容器启动命令与 Dockerfile 中的 ENTRYPOINT 和 CMD 命令关系如下表：

镜像 Entrypoint	镜像 CMD	容器启动命令	容器启动参数	最终执行
[cd]	[/root/test/]	未设置	未设置	[cd /root/test/]
[cd]	[/root/test/]	[mkdir]	未设置	[mkdir]
[cd]	[/root/test/]	[mkdir]	[/opt/test/]	[mkdir /opt/test/]

注意：启动命令必须为容器镜像支持的命令，否则会导致容器启动失败。

4.2.3.2 环境变量

支持手动输入在容器中设置环境变量。环境变量为容器应用提供极大的灵活性，可以在应用中使用环境变量，在创建容器时为环境变量赋值，容器运行时读取环境变量的值，从而做到灵活的配置，而不是每次都重新编写应用程序制作镜像。

4.2.3.3 健康检查

健康检查是指容器运行过程中，根据需要，定时检查容器中应用健康状况。基于 Kubernetes，提供了两种健康检查的方式：存活探针和业务就绪探针。

存活探针（liveness probe），探测容器是否已经启动：该检查方式用于检测容器是否存活，类似于执行 ps 命令检查进程是否存在。如果容器的存活检查的结果为失败，容器实例会对该容器执行重启操作；若容器的存活检查成功则不执

行任何操作。

业务就绪探针（readiness probe），探测容器业务是否已经就绪：该检查方式用于检测容器是否准备好开始处理用户请求。一些程序的启动时间可能很长，比如要加载磁盘数据或者要依赖外部的某个模块启动完成才能提供服务。这时候程序进程在，但是并不能对外提供服务。这种场景下该检查方式就非常适用。

4.2.3.4 生命周期

基于 Kubernetes 提供了容器生命周期钩子，在容器的生命周期的特定阶段执行调用，比如容器在停止前希望执行某项操作，就可以注册相应的钩子函数。

目前生命周期钩子支持：启动后处理（postStart，容器启动后触发），停止前处理（preStop，容器停止前触发）。

4.2.3.5 容器实例 HA

当容器实例所在物理主机故障后，借助高可用机制，可将该容器实例在其他正常的物理主机上启动起来，尽量减少因物理主机故障引起的业务中断。

采用 HA 主调度服务结合宕机事件主动上报机制实现容器实例的高可用服务，HA 主调度服务包含 HA 监听器和 HA 事件处理器，HA 监听器用于循环监听 HA 事件处理的结果，若有一次没处理完毕的宕机事件，HA 监听器则继续进行尝试处理；HA 事件处理器用于接收并处理主动上报的宕机事件 HA 宕机事件主动上报机制，物理主机或容器实例发生宕机事件，并将事件快速主动上报给 HA 主调度服务器的 HA 事件处理器，HA 主调度服务将该服务器上运行状态的容器实例在集群内其它服务器上重新分配资源并进行重新启动。

4.2.3.6 ConfigMap

很多应用在其初始化或运行期间要依赖一些配置信息。大多数时候，存在要调整配置参数所设置的数值的需求。ConfigMap 是 Kubernetes 用来向容器实例 Pod 中注入配置数据的方法。ConfigMap 允许将配置文件与镜像文件分离，以使

容器化的应用程序具有可移植性。

ConfigMap 供容器实例使用的典型用法为：1）作为为容器内的环境变量；2）以 Volume 的形式挂载为容器内部的文件或目录。

支持以 yaml 文件导入生成 ConfigMap，也可以将 ConfigMap 导出为 yaml 文件。

4.2.3.7 事件

Kubernetes 事件（Event）是一种资源对象，用于展示集群内发生的情况。Kubernetes 系统中的各个组件会将运行时发生的各种事件（例如创建 POD 时拉取镜像失败，节点服务异常等）上报给 apiserver。apiserver 将 Event 存储在数据库内，强制执行保留策略：在最后一次的事件发生后，删除 1 小时之前发生的事件。

通过查看容器实例的事件，可以了解容器实例的活动状况，确认容器实例是否正常运行，以及运行异常时的错误原因。

4.2.3.8 镜像拉取策略

容器的镜像拉取策略用 imagePullPolicy 表示，imagePullPolicy 的不同取值会影响 kubelet 尝试拉取（下载）指定的镜像。

以下列表包含了 imagePullPolicy 可以设置的值，以及这些值的含义：

Always：每当 kubelet 启动一个容器时，kubelet 会查询容器的镜像仓库，将名称解析为一个镜像摘要。 如果 kubelet 有一个容器镜像，并且对应的摘要已在本地缓存，kubelet 就会使用其缓存的镜像； 否则，kubelet 就会使用解析后的摘要拉取镜像，并使用该镜像来启动容器。

IfNotPresent，只有当镜像在本地不存在时才会拉取。

Never，Kubelet 不会尝试获取镜像。如果镜像已经以某种方式存在本地，

kubelet 会尝试启动容器；否则，会启动失败。

4.3 存储虚拟化

InCloud Sphere 提供基于主机的存储虚拟化功能。将不同类型的存储设备抽象为逻辑存储资源，以提供统一全面的存储服务。InCloud Sphere 存储管理支持 IP-SAN、FC-SAN、NAS、裸设备存储及本地盘，同时支持外接的分布式存储，经过 FC、iSCSI、FCoE 等不同的适配器与主机连接。接入主机的存储设备经过 InCloud Sphere 存储虚拟化技术对外提供统一的文件或块的存储空间。还提供了精简置备/厚置备延迟置零磁盘、磁盘快照、存储冷热迁移、扩容及克隆等虚拟磁盘相关功能。

另外，InCloud Sphere 还支持 SAN 存储直通技术，提高数据库在云平台上的运行性能。提供了数据存储扩容功能，有效提高数据存储扩展性。

4.3.1 物理存储设备

4.3.1.1 存储适配器

存储适配器为 InCloud Sphere 主机提供了特定存储单元或网络的连接。目前支持的适配器类型有 iSCSI、FC、FCoE 等，实现对 iSCSI 设备，FC 设备以及 FCoE 设备的识别与管理，用户可以灵活的添加或移除磁盘设备。提供的适配器刷新功能，可以在主机连接新的存储设备或断开原有的存储设备后，及时更新主机连接的存储设备。图 4.2.1-1 是 I/O 请求图：

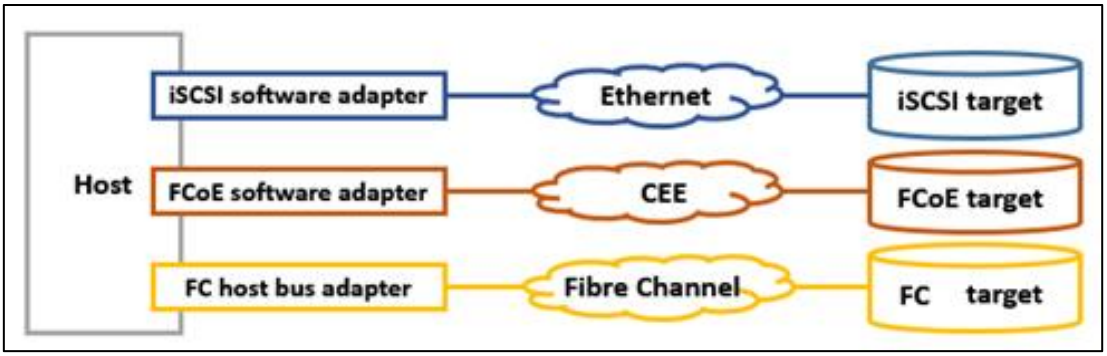


图 4.3.1-1 主机通过适配器与存储设备连接示意图

iSCSI 协议使得 SCSI 命令运行于 TCP/IP 之中，令主机通过 IP 网络与存储设备进行数据传输。InCloud Sphere 支持硬件和软件 iSCSI 存储适配器。

适配器类型	描述	硬件需求	网络需求
硬件 iSCSI	使用专用 iSCSI HBA 卡将主机连接到 IP 网络。	iSCSI HBA 卡	IP 网络
软件 iSCSI	使用标准网卡将主机连接到 IP 网络上的 iSCSI 目标，由 InCloud Sphere 的 iSCSI 软件协议栈对数据进行解析。	普通网卡	IP 网络

FCoE 协议将 FC 协议帧封装到以太网帧中，可以通过 10 Gb 增强型以太网（CEE）进行数据传输，因此主机无需特殊的光纤通道去连接 FC 存储。InCloud Sphere 支持两种类型的 FCoE 适配器：硬件 FCoE 适配器和软件 FCoE 适配器。

适配器类型	描述	硬件需求	网络需求
硬件 FCoE	使用专用 CNA/FCoE 卡将主机连接到 CEE 网络上，支持主机功能卸载。	CNA 卡	增强型以太网
软件 FCoE	使用通用网卡将主机连接到 CEE 网络上，由 InCloud Sphere 相关软件协议栈执行数据解析，主机 CPU 负载较大。	支持数据中心桥接（DCB）功能和 IO 卸载功能的网卡	增强型以太网

业界典型的分布式存储技术主要有分布式文件系统存储、分布式对象存储和分布式块设备存储等几种形式。分布式存储技术及产品日趋成熟，在各行各业得到了广泛使用。分布式存储系统的挑战主要在于数据和状态信息的持久化，要求在自动迁移、自动容错和并发读写的过程中，保证数据的一致性。

通常分布式存储软件具有以下特点：

- 高性能：数据分散存放，实现全局负载均衡，分布式缓存；
- 高可靠：采用集群管理方式，不存在单点故障，灵活配置多数据副本，不同数据副本存放在不同的机架、服务器和硬盘上，单个物理设备故障不影响业务的使用，系统检测到设备故障后可以自动重建数据副本；
- 高扩展：没有集中式存储控制器，支持平滑扩容，容量几乎不受限制；
- 易管理：存储软件直接部署在服务器上，没有单独的存储专用硬件设备。

4.3.1.2 存储多路径

存储多路径通过使用多个物理链路来访问存储设备，通过容错、I/O 流量负载均衡甚至更细粒度的 I/O 调度策略等方式，为存储系统提供更高的可用性和性能优势。InCloud Sphere 提供开放式的多路径模块，支持多个多路径插件同时操作，允许第三方软件开发人员为特定的存储阵列设计自己的负载均衡和故障切换机制。

InCloud Sphere 默认采用内置的多路径模块，用户可以根据业务需要，调整多路径的相关配置。内置存储多路径有两种配置方式，即主从配置方式和双活配置方式。两种不同的配置为用户带来不同的收益。

1）冗余，通过配置多路径为 active/passive（主从）方式，可以提供访问存储的容错能力。如果主路径上的任何设备（光纤、交换机、控制器）发生故障导致所有主路径断开连接，则 I/O 会自动切换到从路径上，使得数据能够继续传输。

2）性能提升，多路径也可以配置为 active/active（双活）方式，此时所有路径都处于工作模式，系统可以自动均衡不同路径上的负载，使得 I/O 性能达到最大化

在连接 iSCSI 设备时，需要指定相应的网络端口，为了提高存储的使用性能，建议配置专门的数据网端口，在配置 iSCSI 多路径时，建议配置不同网段的数据网端口。

第三方多路径插件接入时，需要替换内置多路径模块提供的功能，主要包括：

1）加载和卸载多路径插件。

2）提供逻辑设备和物理路径 I/O 统计信息。

3）处理物理路径发现和移除。

4）处理逻辑设备的 I/O 请求、排除等。

4.3.1.3 物理磁盘

InCloud Sphere 支持对磁盘设备的管理，能方便管理主机上连接的磁盘设备。该磁盘既包括本地磁盘，也包括通过存储适配器连接的外置存储（IP SAN，FC SAN 或 FCoE）所映射的磁盘。

InCloud Sphere 提供支持动态刷新物理磁盘功能，用户可以灵活的添加或移除磁盘设备；能够详细的显示磁盘的相关信息以及各磁盘的属性、路径、I/O 等以及磁盘故障离线等监控信息，实现了存储资源整体可视化。

4.3.1.4 裸磁盘映射

裸磁盘映射通过将 SCSI LUN 或本地硬盘以块设备的方式直接映射到虚拟机中，可以为虚拟机提供接近物理磁盘的性能。当虚拟机对裸磁盘执行读写操作时，虚拟机内的块设备前端驱动将对应的 I/O 操作传递给半虚拟化驱动后端，后端驱动再将这个 I/O 操作作用在实际的物理存储设备，并将其应答返回给虚拟机，为高 I/O 的应用提供支持，保证应用的读写性能，发挥应用的最大优势。虚拟机进行裸磁盘映射时，可选格式化功能，并支持虚拟机裸磁盘映射 HA 功能，保证了裸磁盘映射的虚拟机同样具备 HA 等高级功能。具体流程如图 4.2.1-2 所示。

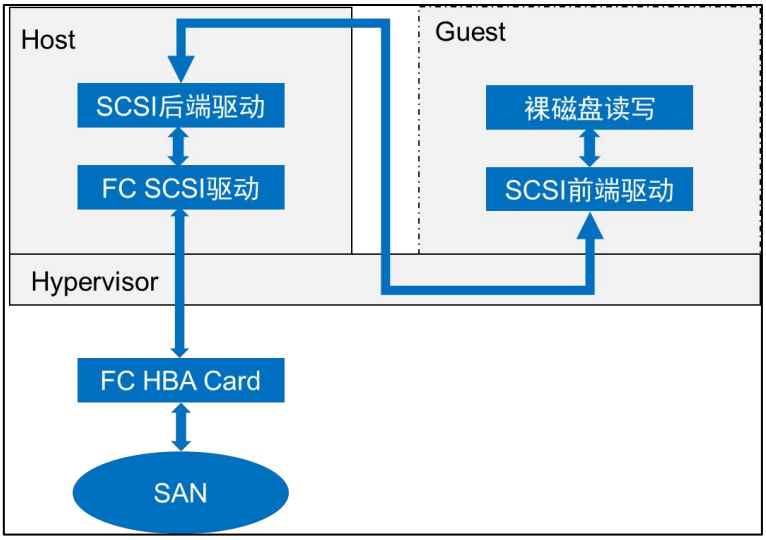


图 4.3.1-2 裸磁盘映射

4.3.1.5 iSER

InCloud Sphere 基于 iSER 实现了更高速率的存储数据传输。iSCSI Extensions for RDMA(iSER)是基于 RDMA 技术的 iSCSI 协议的扩展。iSER 和和标准 iSCSI 的不同点主要是 SCSI 读写命令的执行方式，iSER 的数据传输均是通过 RDMA 读写命令实现。相较于 iSCSI，iSER 可降低传输时延和 CPU 利用率，同时又与 iSCSI 协议兼容，且保留了 iSCSI 协议的诸多高级特性，如安全性、高可用等。

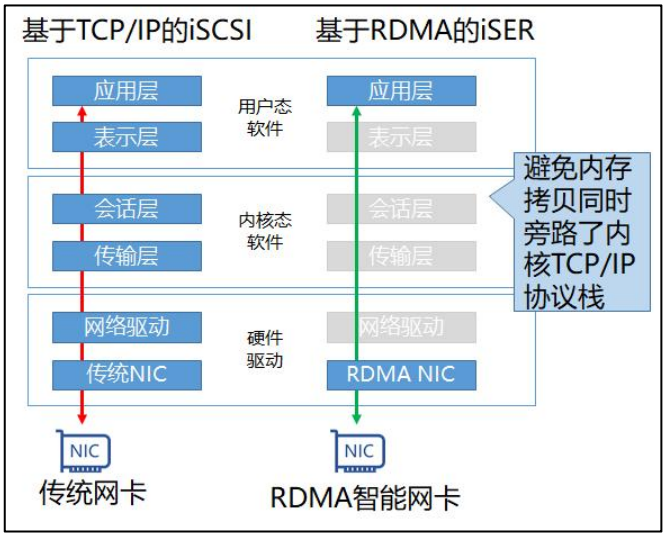


图 4.3.1-3 iSER 与 iSCSI 的比较

4.3.2 虚拟化存储池

4.3.2.1 存储池

由于存储设备的能力、接口协议等差异性，利用 InCloud Sphere 的存储虚拟化技术，可以将不同的存储设备转换为统一管理的存储池，存储池可理解为一种逻辑容器，它以文件或块的形式来存取虚拟机的相关资源，如磁盘镜像、快照和备份等数据。InCloud Sphere 针对用户不同的存储设备和不同的业务需求，提供了多种数据存储类型，包括 NFS、CFS、本地存储池、裸设备存储池等，同时还支持外接分布式存储。

CFS 存储池采用的是集群文件系统，它提供共享和专享两种模式使用模式。共享 CFS 数据存储可以被集群中的多个主机访问使用，为 HA 和 DRS 等高级功能的实现提供了支持；专享数据存储是充分利用 x86 服务器上的本地硬盘资源，格式化完成的数据存储只允许被此服务器主机进行访问。

NFS 存储池是提供访问 NAS 服务器的功能，使用 NFS 数据存储之前需要按照特定的配置、网络和 NFS 数据存储准则。NAS 服务器需要配置主机能访问的网络连接。同时在配置方面，NAS 服务器需要支持 NFS v3 、NFS v4.1 两种协议版本之一，同时支持启动 no_root_squash 权限控制，并确保共享目录添加读/写权限。用户不能使用不同的 NFS 协议版本挂载同一个数据存储，因为 NFS v3 和 NFS v4 客户端不使用相同的锁定协议可能导致用户数据不一致。

4.3.2.2 存储池扩容

当 InCloud Sphere 数据存储的使用量达到一定的阈值时，为了不影响系统的性能，需要向系统动态的增加存储的容量，以保证系统的高可用性，即存储扩容。存储的扩容不影响原数据存储的使用，新添加的数据磁盘可立即作为数据存储直接使用。

为了保证存储扩容的安全性，提供了维护模式方案，用户将存储池进入维护

模式，中断业务，扩容完成后，退出维护模式自动恢复业务。

4.3.2.3 存储池柔性卸载

基于存储池柔性卸载特性，当存储设备无法下发 IO 时，不会引发主机重启，从而保证客户业务在极端情况下不会因为主机重启而中断。当主机连接不到存储设备心跳盘，则该主机会卸载相应的存储池，虚拟机通过 HA 机制在其他可正常访问存储设备的主机上重启。

4.3.2.4 存储域通信模式

共享 CFS 存储池的节点位于同一个存储域中，通过节点间通信完成对共享资源的并发访问控制。存储域支持三种通信模式：网络通信模式、磁盘通信模式和混合通信模式。网络通信模式适用于网络稳定的环境，性能好但对网络波动容忍度差。磁盘通信模式不依赖于网络，但性能要略差于网络通信模式，且对集群规模有一定限制。混合通信模式也称网络/磁盘主备通信模式，优先使用网络通信模式，若遇到网络故障则自动切换到磁盘通信模式。混合通信模式适用于普通网络环境，兼顾网络通信模式性能的同时又能够容忍网络波动，大幅提升 CFS 存储池的稳定性。

4.3.2.5 裸设备存储池

InCloud Sphere 提供了基于小 LUN 的高性能虚拟磁盘技术方案，通过将存储设备上创建的 pool 以 iSCSI 或 FC 的形式挂载给主机，InCloud Sphere 能够创建以 pool 为单位的裸设备存储池。用户可以在裸设备存储池上创建裸设备磁盘，每个裸设备磁盘对应了 pool 里面的一个 LUN。这种类型的虚拟磁盘不需要在存储设备上创建集群文件系统，从而有效降低 I/O 路径性能损耗。

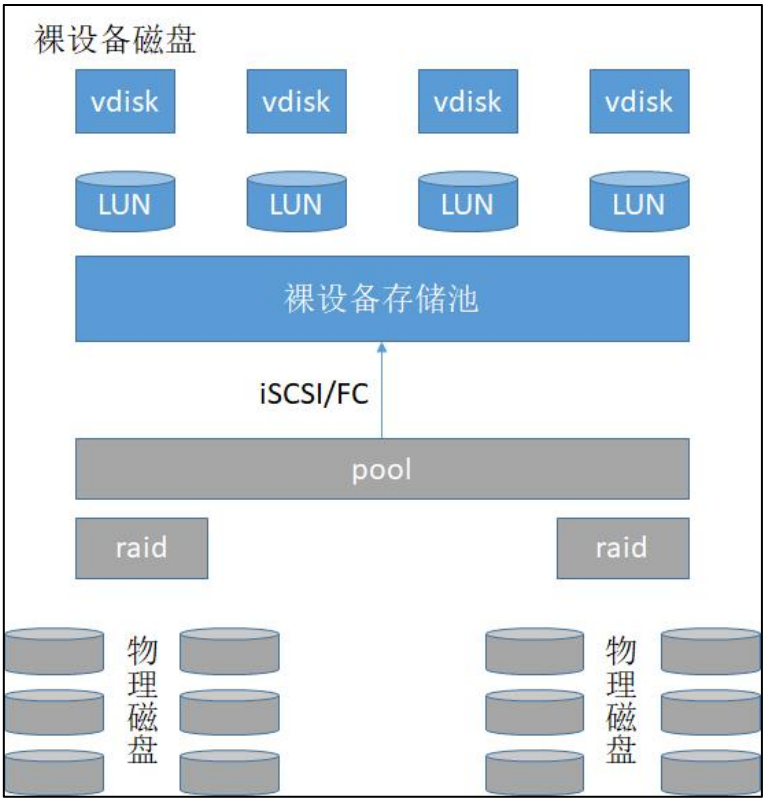


图 4.3.2-1 裸设备磁盘架构

4.3.2.6 双活存储池

双活存储池为虚拟机提供双活磁盘功能，虚拟机同时挂载两个相同的副本磁盘，两个副本磁盘构成一个逻辑上的双活磁盘，两个副本磁盘可以分别存储在SDS和阵列存储上，只要一个副本磁盘可用，即可保证虚拟机业务不中断。

当虚拟机写数据时，同时向两个副本磁盘写入数据，满足强一致性后返回写入成功标志；当虚拟机读数据时，从主副本磁盘读出数据。

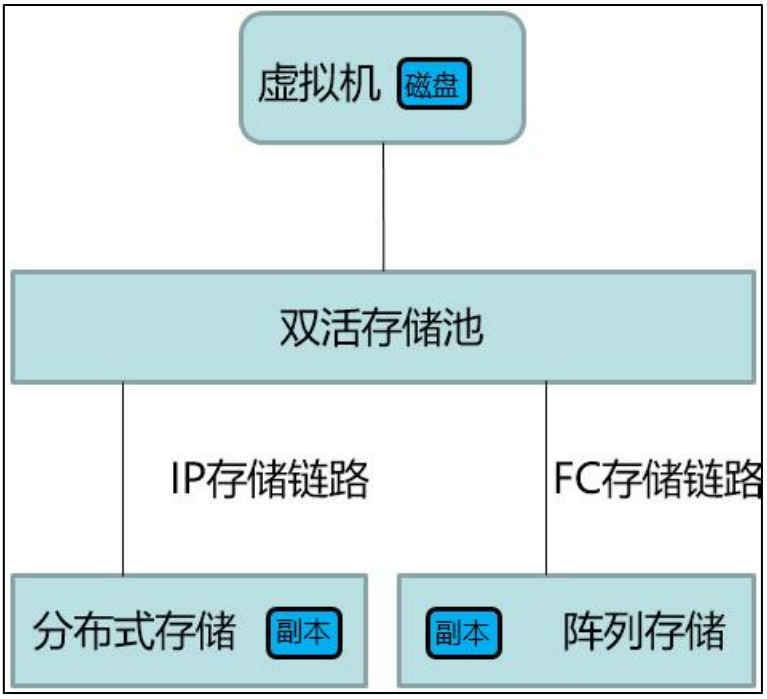


图 4.3.2-2 双活存储池架构

4.3.2.7 存储池加速

传统的存储池内或者存储池间的数据拷贝需要首先将数据拷贝到主机内存中，然后再写入目标存储池。这个过程需要占用 CPU 和内存资源，同时又要占用存储链路带宽，且数据拷贝速率受限于存储链路带宽。

SCSI SPC-3 提出了新的 SCSI 增强指令 XCOPY(Extended COPY)， 可支持存储阵列内的数据拷贝卸载，数据可直接在存储阵列内进行数据拷贝，而无需经由存储链路通过 CPU 进行数据中转。

InCloud Sphere 存储池硬件加速特性基于 XCOPY SCSI 指令，可实现虚拟磁盘文件拷贝卸载，既提高了数据拷贝的速率，又能降低系统的整体开销。此外，对于不支持 XCOPY SCSI 指令的存储阵列，虚拟磁盘文件拷贝可自动识别并转换为传统拷贝方式，以保持系统的兼容性。

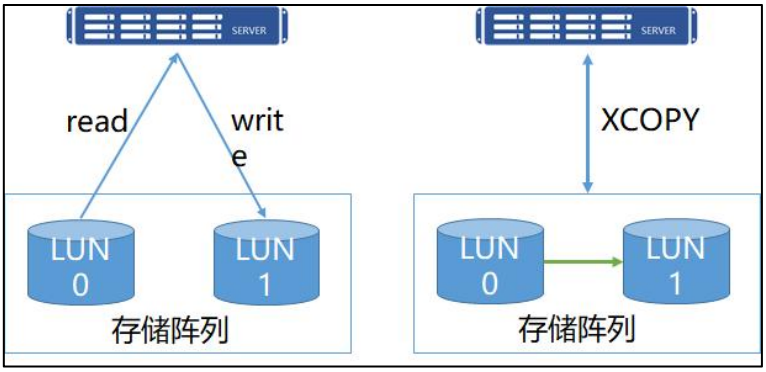


图 4.3.2-3 存储池加速

4.4 网络虚拟化

4.4.1 网络虚拟化架构

InCloud Sphere 集成了 OpenvSwitch 交换机和 OpenStack Neutron 组件，自身实现了 VLAN、VxLAN、安全组、分布式路由器、分布式 NAT 网关、分布式防火墙、DHCP 等丰富的网络功能；同时借助 Neutron 的开放性，实现 InCloud Sphere 网络与第三方 SDN 控制器的标准化对接。

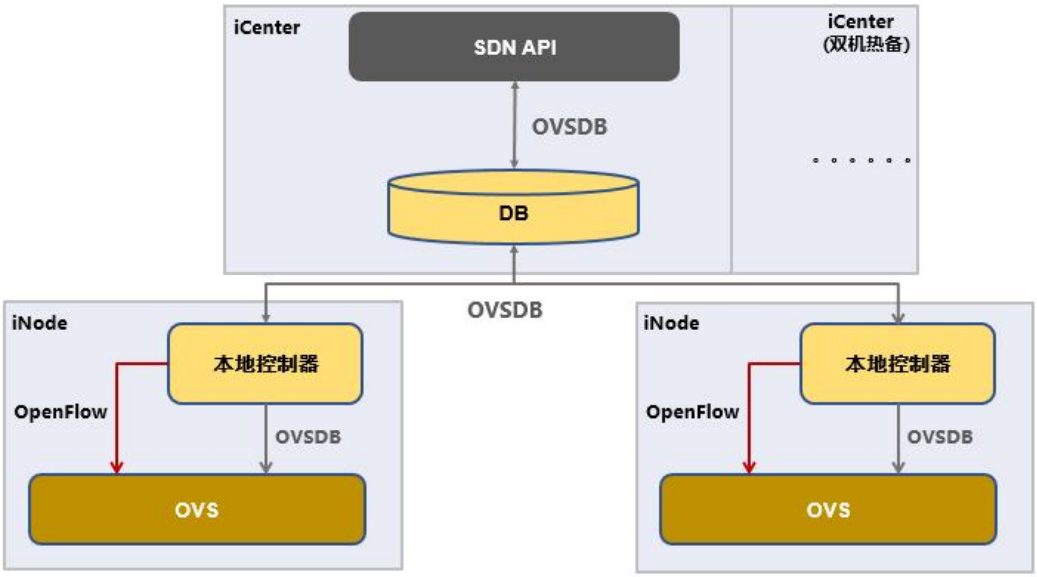


图 4.4.1-1 网络虚拟化架构

4.4.2 网络转发

4.4.2.1 SDN 网络

InCloud Sphere 同时支持基础网络和 SDN 网络，基础网络支持传统的二层 VLAN 网络转发。SDN 网络包含 VLAN 和 VxLAN 两种网络，支持更丰富的网络功能。VLAN 网络支持 801.1Q 协议，一个 VLAN 属于一个二层广播域，在同一个 VLAN 里的虚拟机都是二层可达的。VxLAN 网络是基于隧道技术的 overlay 网络，VxLAN 网络中的数据包通过 VTEP 网络端口封装成 UDP 包进行传输，二层的包通过封装在三层传输，克服 VLAN 和物理网络基础设施的限制。

4.4.2.2 分布式路由器

提供三层转发功能，运行在各个计算节点，不同子网之间的数据流量通过路由器转发。可以提供 SNAT 功能，实现虚拟机对外部网络的访问，不同计算节点的虚拟机使用 SNAT 出外网时，流量集中在一个计算节点访问外网。同一个计算节点不同子网的虚拟机间互访时，流量在本地转发，无需出计算节点。

4.4.2.3 静态路由

基于分布式路由器，提供静态路由功能，可以根据网络流量转发需要，添加多条静态路由。使用最长匹配方式查找静态路由转发表，并支持 ECMP(Equal Cost Multi-Path)等价路由，可以对流量进行负载均衡。

4.4.2.4 策略路由

策略路由是基于目标网络进行路由的、更加灵活的数据包路由转发机制，通过创建策略路由，您可以实现管理网、数据网、业务网流量的分离，举例来说，用户可以为数据网端口配置策略路由，指定数据网络流量的转发路径，在这里，策略路由需要指定一个目的地址，这样去往这个目的地址的流量将只会流经此数据网端口，这样做可以做到数据网流量与管理网、计算网流量的隔离，这样可以提高网络安全性，避免网络之间的相互干扰。

4.4.2.5 管理网多 VLAN

管理网络是存在于管理网络交换机上的一种网络。管理网络交换机在管理节点部署完成时默认创建，并默认创建管理网络。用户可以在管理界面中看到默认创建的管理网络交换机和管理网络，但是没有关联主机，用户添加主机之后，系统将管理网络交换机与添加的主机进行关联，并在管理网络交换机上创建默认的管理网络端口。

默认创建的管理网络的 VLAN 由用户在部署安装系统时配置，管理网络支持配置的 VLAN 范围为 0-4094；在一个 iCenter 集群下，InCloud Sphere 支持管理网多 VLAN 部署。大规模环境下，如果需要对不同的主机划分不同的管理 vlan 网络，InCloud Sphere 能够较好的满足要求。

4.4.2.6 组播

普通虚拟交换机支持开启组播功能，可以在虚拟交换机的主机列表中开启、关闭、批量开启、批量关闭组播。提供虚拟交换机的转发效率。源主机向多点目标主机发送信息需求时，源主机只发送一份数据，数据的目的地址是组播组地址，这样，凡是属于该组的成员，都可以接收到一份源主机发送的数据的拷贝，它提高了数据传送效率，解决了单播和广播方式效率低的问题。

4.4.2.7 IP 网络

IP 网为虚拟机的迁移、备份等功能提供网络服务。用户可以根据用途将 IP 网标记为迁移网络、备份网络及其他网络类型。默认情况下，虚拟机在迁移、备份时会产生的流量是由管理网承载，在执行虚拟机迁移、备份任务时，可能会因管理网的流量过大，影响系统使用。使用 IP 网承载上述流量，实现了迁移、备份等流量与管理网隔离，减轻管理网压力，增强系统稳定性，提升用户操控体验。

4.4.2.8 上行链路主备切换

当虚拟交换机上行链路绑定模式为主备模式时，可在线对上行链路网卡进行主备切换。使原备网卡切换为主网卡，原主网卡变成备网卡。

4.4.2.9 上行链路 ARP 广播模式

在去堆叠网络架构中，为了提高可靠性，一般需要在核心与接入 TOR 交换机建立 ECMP 等价路由。ECMP 等价路由的建立依赖虚拟交换机上行链路同时转发 ARP 报文到两个接入交换机。ARP 广播模式在配置普通虚拟交换机上行链路时开启，其只能工作在 Ovs-bond 的 LACP 模式下。ARP 广播模式通过复制 ARP 报文并分别上送给上行链路聚合口的每个成员，实现“ARP 双发”，能够灵活适配去堆叠的组网方式，提高网络可靠性。

注意：ARP 广播模式需要与物理网络拓扑的接入交换机及核心路由器等网络设备配合使用，请谨慎启用该功能。

4.4.3 网络服务

4.4.3.1 端口转发

端口转发是一种 NAPT（网络端口地址转换）技术，可以节约公网 IP，可以基于 SNAT 地址或者浮动 IP 实现。NAT 和 NAPT 的区别是 NAT 是 IP 地址的转换，NAPT 是 IP 地址+端口的转换，使<内部地址+内部端口>与<外部地址+外部端口>进行转换，用于只有一个公网 IP 但是有多个业务系统需要被互联网访问的场景。

4.4.3.2 负载均衡

浪潮云应用交付系统 InCloud SSA 采用负载均衡、应用加速、流量控制等技术，提升了核心业务系统的可用性、效率和安全性。InCloud SSA 能够支持四到七层应用负载均衡功能，提供加权轮训、节点最少链接、源 IP 哈希等多种算法的全局负载均衡，采用业内领先的内存缓存技术，有效降低服务器负载，并提升访问响应速度。

通过将 InCloud SSA 作为虚拟机部署在 InCloud Sphere 系统之上，InCloud Sphere 为业务虚拟机提供了统一的负载均衡管理平台，如图 4.3.3-1，虚拟机通

过业务网（黄色线路）连接 SSA 虚拟机，SSA 虚拟机通过浮动 IP（红色线路）向外部提供服务；当外部客户机发起业务请求时，SSA 基于负载均衡算法选择 Server 端业务虚拟机提供服务。

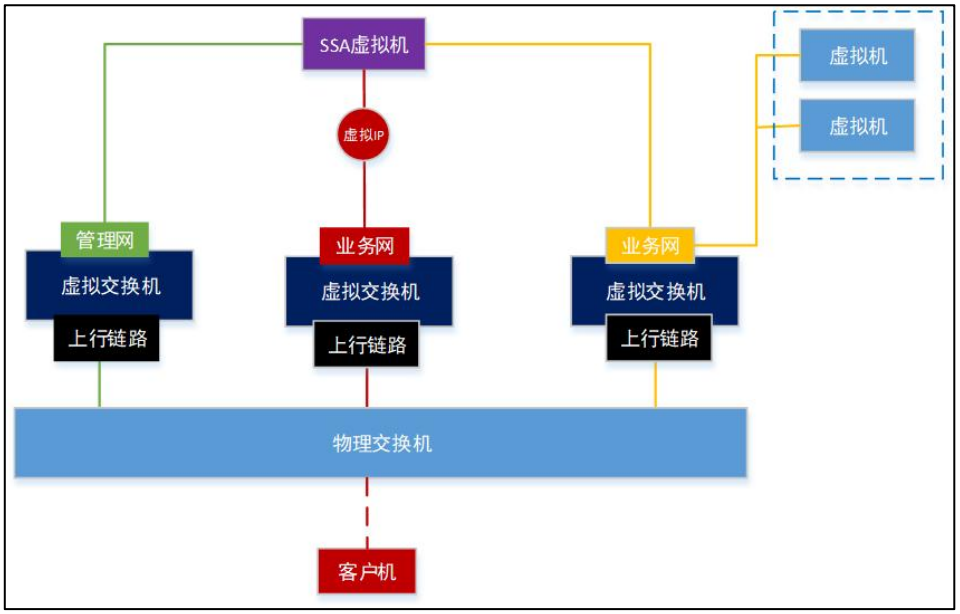


图 4.4.3-1 负载均衡网络部署示例

4.4.3.3 分布式 NAT 网关

提供从外部网络直接访问内网虚拟机的功能，实现了 SNAT 和 DNAT 功能，建立外网 IP 和虚拟机 IP 的一对一映射。当外网访问该虚拟机时，可以直接访问虚拟机所绑定的 NAT 地址。外部网络访问虚拟机时，流量直接从虚拟机所属计算节点访问虚拟机，无需集中式网关。

4.4.3.4 分布式 DNS 服务器

分布式 DNS 服务器为数据中心下的 SDN 网络提供了域名解析服务，可设置域名与 IP 的映射关系，相比较于传统 DNS 服务器避免了单点故障，分布式 DNS 服务器相当于在每个计算节点运行了一个联动的 DNS 服务器服务，提供了更加可靠的分布式 DNS 域名解析服务，任何一个计算节点宕机均不会影响域名解析服务。当一个域名对应多个 IP 地址时，分布式 DNS 服务器还会对不同来源的解

析请求，分别给予不同的解析结果，使被访问的目标服务负载更加均衡化。

4.4.4 网络安全

4.4.4.1 DHCP 防护

DHCP 防护能够保障合法的 DHCP 服务器的正常工作，对虚拟机非法 DHCP 服务器发出的报文进行阻断，保障虚拟机能够从合法 DHCP 服务器正常获取 IP 地址。

4.4.4.2 广播抑制

在二层网络中，广播包会从除接收端口外的所有端口转发出去，由于这种转发机制，大量的广播包会影响网络性能。广播抑制功能可以设置虚拟交换机允许通过的广播报文带宽以限制虚拟机发送大量的广播报文，防止广播报文攻击。

4.4.4.3 安全组

InCloud Sphere 基于安全组实现相互信任的虚拟机之间通信，绑定同一个安全组的虚拟机使用相同的安全策略，安全组通过规则限制虚拟机的流量收发。虚拟机通过 tap 设备连接到 OpenvSwitch 网桥，OpenvSwitch 网桥通过 Openflow 协议实现安全组功能。

4.4.4.4 分布式防火墙

采用动态可扩展的结构体系，为网络边界、内部网络和子网之间的流量提供多层次、多协议、全方位的安全控制。通过设置防火墙规则，对进出防火墙的流量进行过滤。

4.4.4.5 二层引流

二层引流功能可使 SDN 某个二层网络下所有流量全部引至引流网元，流量经网元处理后再重新注入 SDN 网络，如下图所示。二层引流功能可以与第三方厂商的 NFV 网元对接，如 IPS、防火墙、WAF、VPN 等。

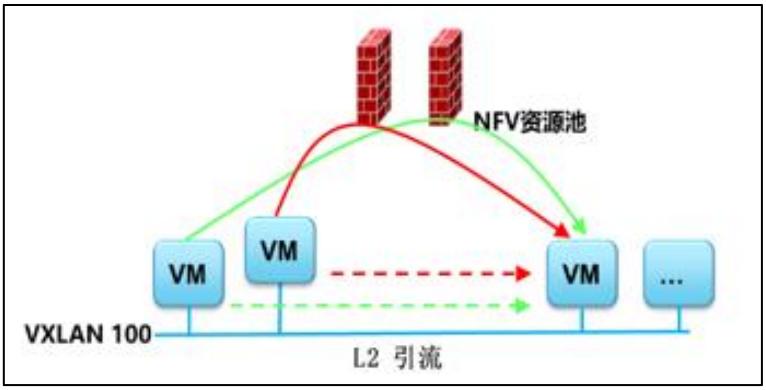


图 4.4.4-1 L2 引流

4.4.4.6 三层引流

与二层引流功能类似，SDN 分布式路由器的三层引流功能也是构建 ICS 生态的关键技术之一，利用三层引流功能可以通过匹配数据包五元组，将流经路由器的三层转发流量，引流到第三方的 NFV 网元设备中，经网元设备处理之后再作正常的三层转发。

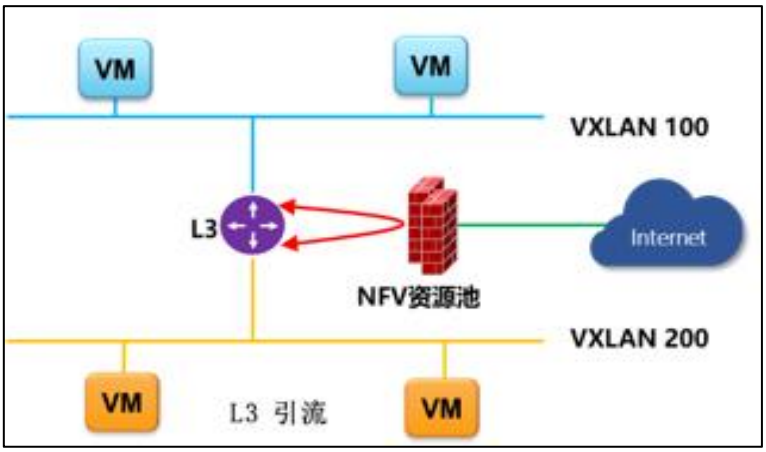


图 4.4.4-2 L3 引流

4.4.5 流量限速

4.4.5.1 业务网 QoS

InCloud Sphere 提供多层次的 QoS 设置方式，业务网设置 QoS 为以 VLAN 网络为单位为加入同一网络的虚拟机设置相同的 QoS，限制一组虚拟机的上行和下行传输速率。

4.4.5.2 虚拟机迁移带宽百分比

通过设置虚拟机迁移带宽百分比可调整虚拟机迁移时所占用的管理网带宽，根据上行链路网卡的协商速率配置虚拟机迁移时占用的带宽，提升虚拟机迁移的速率。

4.4.5.3 网络优先级

网络优先级功能，在上行链路中，通过报文中的 MAC 地址，标识流量来源，根据不同的规则应用不同的限流带宽。使不同的虚拟网卡对外通信时的带宽也不同。从而实现网络 IO 优先级配置。

虚拟机网卡开启了网络优先级时，可以选择低/中/高进行配置，优先级越高所分配的网络带宽越高。

4.4.5.4 无损网络 PFC

PFC（Priority-based Flow Control，基于优先级的流量控制）是构建无损以太网的必选手段之一，能够逐跳提供基于优先级的流量控制。设备进行报文转发时，根据报文中的优先级映射到对应的队列进行调度转发。当某一优先级报文的发送速率大于接收速率，导致发生拥塞时，下游设备向上游设备发送 PFC PAUSE 帧，上游设备收到后停止该优先级的报文发送。当上游设备收到 PFC XON 帧后或者经过一定的老化时间后恢复报文的发送。使用 PFC 功能，可以基于优先级对流量进行控制，达到相同链路上不同类型的报文互不影响的效果，并且实现报文的零丢包传输。

在存储交换机的配置中开启 PFC 功能时，可以选择使能 PFC 优先级，范围 0~7。根据使用场景使能 PFC 优先级后，可在数据网中实现不同类型流量转发互不影响的效果，并且实现报文的零丢包传输。

4.4.6 IP 地址管理

4.4.6.1 业务网 DHCP

虚拟交换机增加业务网络可以选择开启 DHCP，配置网络地址。当虚拟网卡选择所绑定的业务网，开启 DHCP 时，虚拟机开机时，系统会为虚拟网卡自动分配一个 IP 地址。

4.4.6.2 子网

是一个 IPv4 或者 IPv6 地址段，虚拟机的 IP 地址从子网中分配。当子网开启 DHCP 功能时，所有虚拟机的 IP 地址由本地控制器直接分配，报文无需出计算节点。

4.4.6.3 地址分配池

地址分配池可以通过配置地址池的起始地址、结束地址对 SDN 子网网段、分布式 NAT 网关 NAT 资源池进行更加精细的控制。

4.4.6.4 DHCP 方式设置 DNS

InCloud Sphere 支持通过 DHCP 方式设置 DNS，开启 DHCP 的业务网配置了 DNS 后，关联此网络的虚拟机开机后自动完成 DNS 配置，不需要手动设置，为虚拟机设置 DNS 提供了一种方便快捷的手段。

4.4.7 网络加速

4.4.7.1 SDN 硬件加速

SDN 硬件加速功能利用普通网卡的硬件特性，加速 Overlay 网络封装，提高网络转发性能，无需特殊网卡硬件支持。SDN 硬件加速可以充分利用网卡的 TSO/LRO 特性，把分段/合并功能和 TCP 校检转移到网卡上去实现，加速 Overlay 隧道报文的处理，达到提升转发性能的效果。

4.4.7.2 SDN 智能网卡加速

随着软件定义网络（SDN）、网络功能虚拟化（NFV）以及多云管理的快速

发展，服务器网络带宽正面临快速增长，对网络性能加速提出了更高的要求。

VXLAN 等重叠隧道协议和 OpenFlow、Open vSwitch（OVS）等虚拟交换技术的引入使得网络中数据平面的复杂性急剧增加。

数据中心超融合基础架构带来的带宽需求增加使得网络处理功能在 CPU 上产生更大的负载，影响应用程序执行。

传统网卡固定功能的流量处理功能无法适应 SDN 和 NFV。

目前最有效的硬件加速方法为智能网卡，将虚拟交换机功能完全从服务器 CPU 转移到网卡，释放昂贵的服务器 CPU 的计算能力以返回给应用程序，从而更好地扩展网卡功能并提供更高的性能，为边缘计算提供更高的算力。

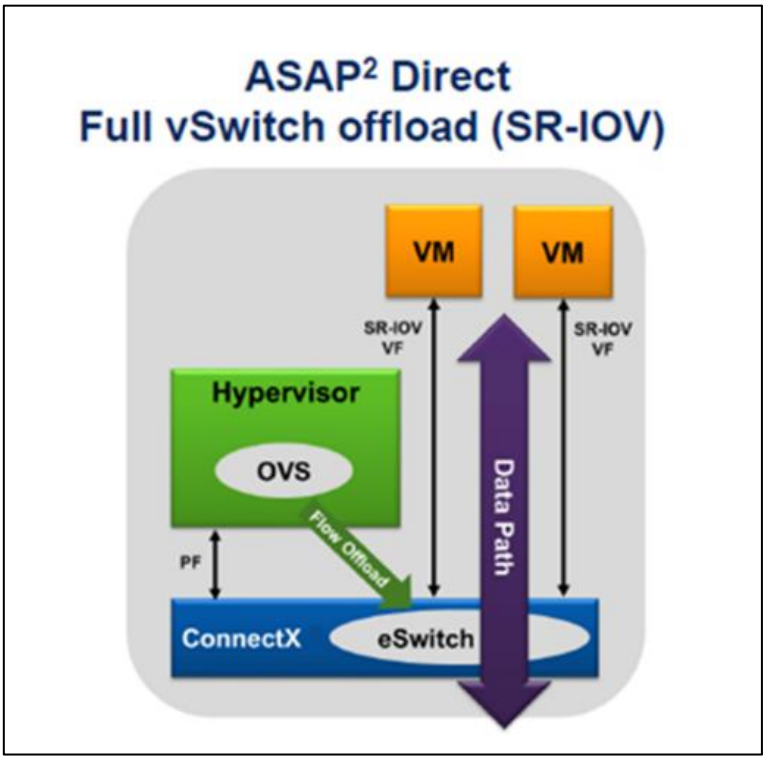


图 4.4.7-1 ASAP 卸载

如上图所示，SDN 智能网卡卸载分布式 SDN 网络的数据平面到智能网卡，通过对虚拟机网卡东西向 Overlay 网络流量和南北向流量卸载，大幅提高网络转发性能并节省主机 CPU 资源。

4.4.7.3 DPDK 网络加速

DPDK 全称 Data Plane Development Kit（数据平面开发套件），是一个开源的数据平面开发工具集，提供了一个用户空间下的高效数据包处理函数库，它通过 EAL 环境抽象层旁路内核协议栈、轮训模式的报文无中断收发、无锁环形内存管理、优化内存/缓冲区/队列管理、基于网卡多队列和流识别的负载均衡、大页内存、NUMA、CPU 隔离和绑定等多项技术，实现了在 X86、ARM、powerpc 处理器架构下的高性能报文转发能力。

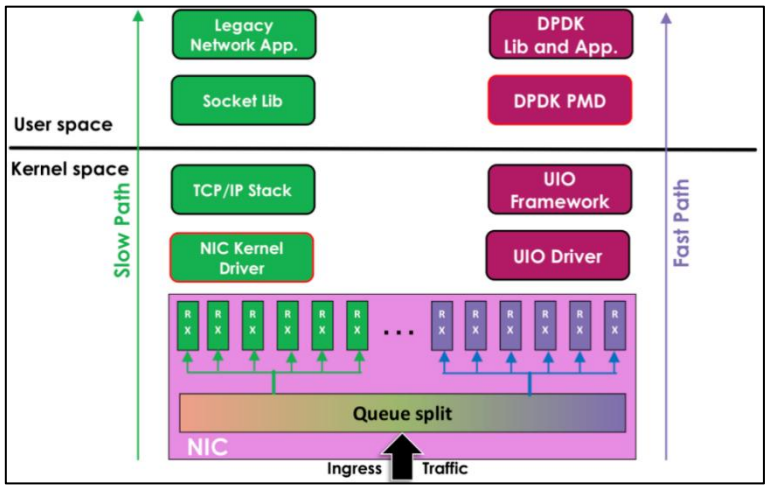


图 4.4.7-1 DPDK 旁路技术原理

通过 UIO（用户态 IO）机制，用户态驱动程序通过 UIO 暴露的/dev/uioX 读取硬件中断，通过 mmap 与外设网卡共享内存，实现与网卡的通讯。使用 UIO 可以通过 read 感知中断，通过 mmap 实现和网卡的通讯。UIO 对用户层屏蔽中断，在用户态采用主动轮询的方式（PMD，Poll Mode Driver）。DPDK 从而可以在用户态做收发包处理，带来 Zero Copy、无系统调用的好处，同步处理减少上下文切换带来的 Cache Miss。Interrupt DPDK 模式解决了网络空闲时 CPU 长期空转（不断的主动轮训）带来的能耗问题。

DPDK 的上层用户态由很多库组成,主要包括核心部件库(Core Libraries)、平台相关模块(Platform)、网卡轮询模式驱动模块(PMD-Natives& Virtual)、QoS

库、报文转发分类算法(Classify)等几大类,用户应用程序可以使用这些库进行二次开发。

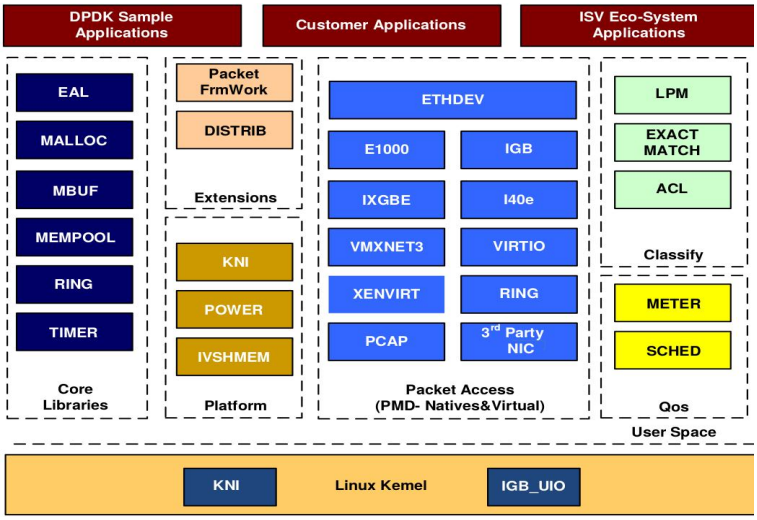


图 4.4.7-2 DPDK 库组件

InCloud Sphere 中的用户态交换机（创建虚拟交换机时启用 dpdk）使用的 ovs-dpdk 即是 ovs 使用 DPDK 二次开发而成，ovs-dpdk 显著改善了 OpenvSwitch 的转发性能，同时保留了其核心功能。它使能数据包由主机物理网卡到虚拟机应用直接转发，几乎完全在用户空间完成。

InCloud Sphere 用户态交换机使能用户空间应用直接由物理网卡获取数据包并处理，一定场景下可以线速处理，虚拟机网络 IO 的能力极大提升。

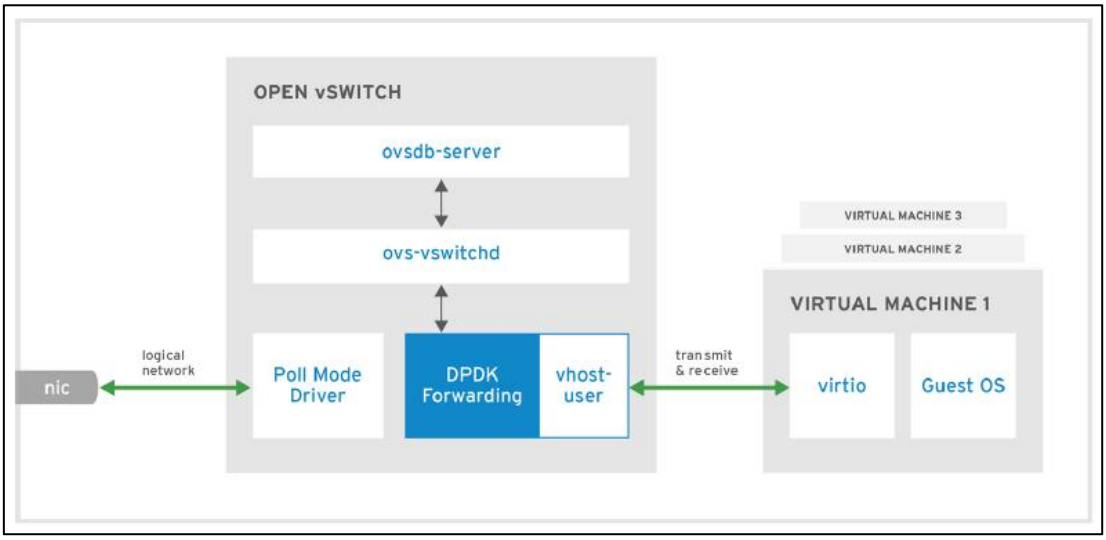


图 4.4.7-3 ovs-dpdk

4.4.7.4 DPDK 上行链路网卡多队列

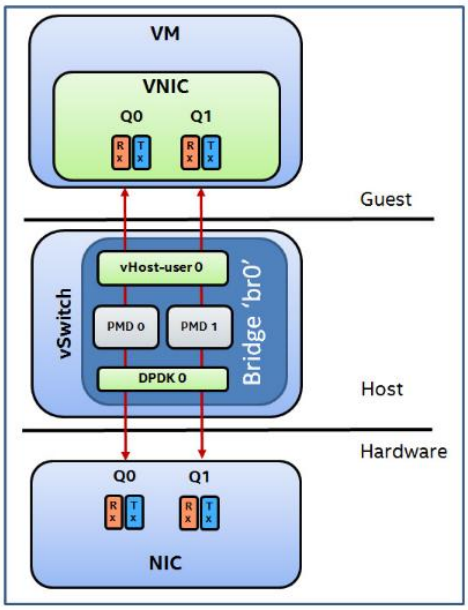


图 4.4.7-4 DPDK 网卡多队列

InCloud Sphere 支持 OVS-DPDK 特性，虚拟交换机（OVS）通过开启 DPDK 可提升虚拟机网络转发性能。

OVS-DPDK 具有一个 DPDK 端口，用于从物理网卡发送和接收流量。默认情况下物理网卡配置有一个队列（Q0），队列由接收（rx）和传输（tx）路径组成，由单个队列发送和接收流量可能会成为网络瓶颈。默认情况下，DPDK 还具有单个轮询模式驱动程序线程（PMD 0），该线程负责在 DPDK 端口上执行发送和接收操作。

上行链路网卡多队列通过在物理网卡开启多个队列，将 DPDK 的多个 PMD 线程的 CPU 和物理的多个队列进行绑定，充分利用多核并行处理特性，提高 DPDK 的转发性能。

4.4.7.5 VIRTIO 虚拟网卡队列长度

Virtio 是一种利用半虚拟化技术提供的 I/O 性能框架。Virtio 网卡驱动包括

前端驱动和后端驱动，前端驱动指安装在 Guest 中的驱动，后端驱动指安装在 Host 中的驱动。Guest 收发包都要经过前端驱动和后端驱动。

在前端驱动和后端驱动上配置一段共享内存，共享内存使用队列结构，队列中的成员指向报文的消息内容。Guest 发送报文时，将报文在内存中的地址信息放入共享内存队列中，然后通过中断的方式通知后端驱动，后端驱动感知到之后，从共享内存中取出报文，然后调用 Host 的物理网卡驱动将报文发出去。Guest 接收报文与上述发送过程相反，流程类似。

共享内存使用队列的组织形式，队列中的每一个或者多个（报文分片）成员指向一个报文。目前 virtio 网卡队列长度支持设置 256，512，1024，2048 和 4096。适当调大队列长度能够提高 Guest 的网络的性能。

4.4.8 流量监控

4.4.8.1 网络流量监控与统计

sFlow（Sampled Flow）是一种基于报文采样的网络流量监控技术，主要用于对网络流量进行统计分析。sFlow 系统包含一个嵌入在设备中的 sFlow Agent 和远端的 sFlow Collector。其中，sFlow Agent 通过 sFlow 采样获取本设备上的接口统计信息和数据信息，将信息封装成 sFlow 报文， 当 sFlow 报文缓冲区满或是在 sFlow 报文缓存时间（缓存时间为 1 秒）超时后，sFlow Agent 会将 sFlow 报文发送到指定的 sFlow Collector。sFlow Collector 对 sFlow 报文进行分析，并显示分析结果。系统支持在普通交换机上启用 sFlow，当启用交换机的 sFlow 后，可以查看交换机上每一个端口的流量监控。

NetFlow 是思科公司作为一项专有技术最先开始研发的，是一种基于网络流信息的统计与发布技术，它可以对网络中的通信量和资源使用情况进行统计和发布，向网络管理人员提供访问通讯量详细信息的手段。一个 NetFlow 系统包括三个主要部分：探测器，监视器，报告系统。探测器是用来监听网络数据的。监视

器是用来收集探测器传来的数据的。报告系统是用来从监视器收集到的数据产生易读的报告的。

普通虚拟交换机开启 NetFlow 可以获取虚拟交换机流量数据，通过 NetFlow 监视器实现更丰富的网络流量分析功能。

4.4.8.2 端口流量镜像

InCloud Sphere 端口流量镜像功能，是用来将源端口或源 VLAN 的网络流量镜像到目的端口，以实现对指定端口或指定 VLAN 网络监听的目的。根据其应用范围不同，端口流量镜像功能主要分为以下三类：

一、本地端口镜像，用来将一个或者多个端口的流量镜像到目的端口。适用于源端口和目的端口位于同一个主机内，且位于同一个虚拟交换机上的场景；

二、本地业务网络镜像，用来将指定 VLAN 的网络流量镜像到目的端口。适用于同一个主机内，且指定的 VLAN 网络和目的端口都位于同一个交换机上的场景，可选的网络范围为 1-4094 的 VLAN 网络；

三、远程端口镜像，用来将一个或者多个端口的网络流量，通过 GRE 隧道镜像到某个远程 IP 地址，适用于目的端口不在本地的场景。

4.4.9 主机管理网 IP 冲突防御

当开启主机管理网 IP 冲突防御后，集群里的主机管理网 IP 受到保护，当其他设备的 IP 与主机的管理网 IP、浮动 IP 冲突时，iCenter 自动识别并屏蔽其他设备，不影响集群内各节点之间的正常通信，并以告警的形式提醒用户及时处理冲突。

ARP 报文是使用 IPv4 地址进行通信的前提，两节点之间的通信都是以 ARP 报文开始的。主机管理网 IP 冲突防御正是利用此特性设计的功能。当开启主机管理网 IP 冲突防御功能后，数据中心的各主机管理网 IP 及 MAC 地址保持实时同步，使每台主机都有数据中心内所有主机的 IP 地址与 MAC 地址的映射关系

（包括管理节点的浮动 IP 地址），从而使之具备对所有主机发出 ARP 报文的鉴别能力。

本功能生效之后，每台主机的管理网虚拟交换机，都可以通过流表匹配本数据中心下所有主机 IP 地址相关的 ARP 报文，并对这些报文进行识别。当这些 ARP 报文来自本集群内的主机时，进行正常转发；若报文来自第三方的设备，则将报文丢弃，以免本主机学习到此 IP 对应的错误 MAC 地址，造成与该 IP 对应主机间的通信故障或主机掉线。此外，当主机发现有第三方的设备与自己 IP 发生冲突时，系统还会以告警的方式提醒用户及时处理 IP 冲突，如告警之后的 10 分钟内再无此报文，则自动消除告警消息，若还有冲突的 IP 报文，则告警会一直保持，直到 10 分钟的间隔内检测不到冲突 IP 的 ARP 报文。

4.4.10 主机物理网卡定位

主机物理网卡定位 60s 功能开启后，服务器后面的网卡会闪烁，以便确定对应哪个实际物理网卡。

4.4.11 SDN 拓扑所画即所得

SDN 所画即所得为 SDN 方案中一项特色功能，通过此功能用户可以通过拖拽网元的方式灵活绘制 SDN 拓扑图，使用户可以更快速、直观的方式部署 SDN 业务，用户不需要关心底层的逻辑实现用户可以以可视化的方式维护 SDN 拓扑图。

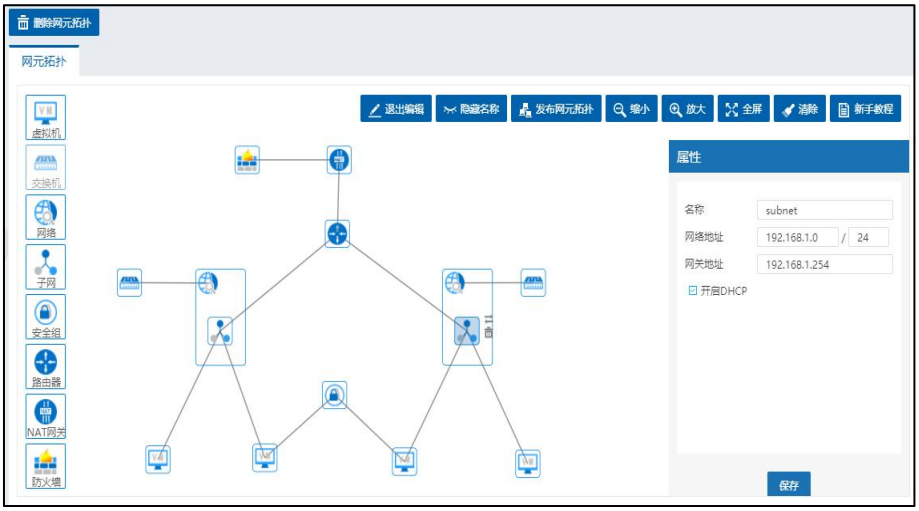


图 4.4.10-1 所画即所得

如上图所示，用户增加网元拓扑图后，在拓扑图上通过简单的拖拽网元图标和连线，即可进行 SDN 业务的部署，以所画即所得的方式完成 SDN 网元拓扑的发布。

4.4.12 物理交换机纳管

物理交换机纳管功能，用户可通过 iCenter 对物理交换机进行可视化配置，直观了解交换机的配置信息和健康状况。可以对交换机进行端口设置和聚合设置，包括设置环路检测、access/trunk 模式、端口速率、开启 LLDP 功能、端口聚合等；可以直观获取交换机型号、系统版本、电源、温度等基本信息；可以查看端口的连接状态、速率、带宽利用率、对端连接设备、光模块类型和匹配状态等信息。

4.5 裸金属服务器

InCloud Sphere 内置裸金属服务器管理功能，支持裸机的创建、删除、开启、关闭等功能，支持通过 qcow2 文件创建裸机，支持在管理界面登录裸机控制台，实现对虚拟机服务和裸金属服务在同一界面下的统一管理。

InCloud Sphere 裸机管理功能基于 Ironic 实现，包含 Ironic API 和 Ironic

Conductor 进程。Ironic Conductor 完成 Ironic 服务的绝大部分工作，其通过 Ironic API 对外开放其功能，Ironic Conductor 与 Ironic API 通过 RPC 进行交互。Ironic 部署裸机过程中需要 deploy image 和 user image 两套镜像，deploy image 用于部署过程的管理，user image 是用户使用的系统镜像。裸机启动后，通过 DHCP 获得 Ironic Conductor 的地址并尝试通过 tftp 协议从 Conductor 获取镜像，Conductor 将 deploy image 部署到物理节点上后，通过 iSCSI 协议将物理节点的硬盘暴露出来，随后通过 qemu 将 user image convert 到物理磁盘上，完成用户镜像部署。

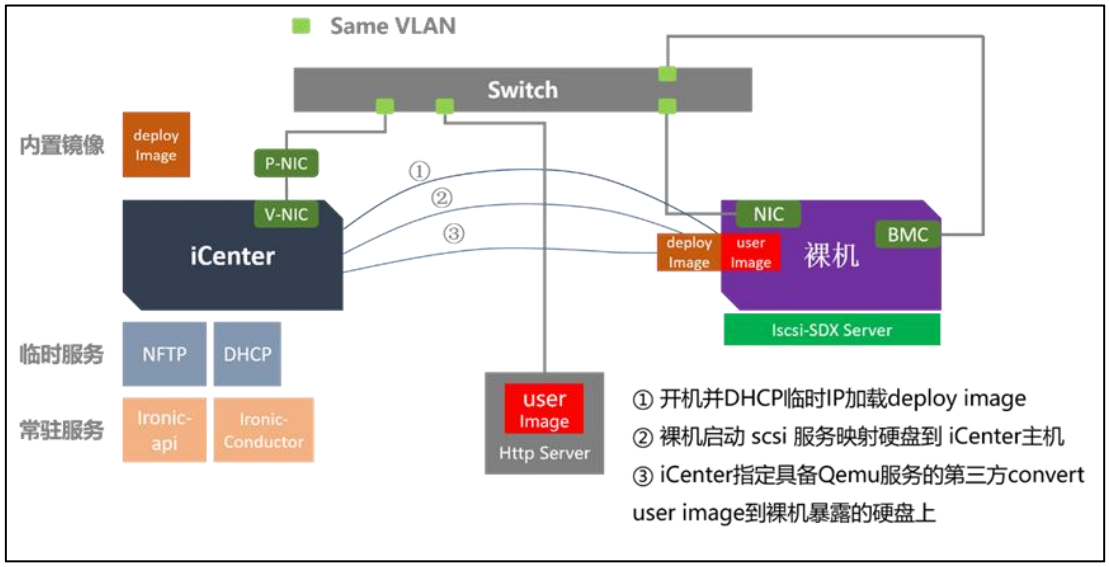


图 4.5-1 裸金属管理技术架构

5 管理与运维

5.1 资源监控

InCloud Sphere 会提供详细的性能指标监视数据，包括 CPU、内存、磁盘、网络信息，会针对每个主机和每个 VM 提供这些指标，同时为存储提供读写速率指标，为交换机提供流量监控指标。这些指标可以直接获取，也可以在 iCenter 或其他第三方应用程序中以图形方式进行访问和查看。

InCloud Sphere 也提供系统和性能警报。警报是指为响应选定系统事件而发生的通知，或在 CPU、内存使用率、网络吞吐量、存储吞吐量或 VM 磁盘活动超过托管主机、VM 或存储库上指定的阈值时发生的通知。可以使用 iCenter 配置这些设置，根据任何可用的主机或 VM 性能指标创建通知。

5.1.1 阈值告警

管理员可以在 iCenter 设置监控阈值，监视其 InCloud Sphere 主机、虚拟机(VM)和存储的性能，并可以多层次的设置阈值。

InCloud Sphere 系统会根据管理员设置的阈值实时监控系统各项指标，当系统性能数据达到阈值条件时，系统会发出告警信息，在 iCenter 页面以弹框形式告知客户，并在页面右侧栏显示告警基本信息；在对象的管理页面也会有告警列表展示；如果提前配置好邮件，系统会把告警信息以邮件形式发送到客户邮箱。邮箱告警支持 SSL/TLS 与 STARTTLS 安全加密协议。

InCloud Sphere 将告警信息分为当前告警与历史告警。已确认并标记为修复告警信息、自动修复的告警信息自动移入历史告警信息。通过告警列表中的高级选项，用户可以根据告警的确认状态、告警级别、告警类型、告警信息关键字及告警时间段进行搜索并导出告警列表。

5.1.2 告警通知

告警信息可以通过多种方式主动推送给其他平台，包括邮件、短信、钉钉、企业微信、SNMP、Microsoft Teams、webhook 等，满足用户多样性的需求。

短信通知方式支持乐讯通和自定义两种方式，使用短信通知功能需要配置对应短信平台的账号、密码、模板、手机号、URL 等参数，当环境产生新告警时，ICS 后台同步触发告警通知逻辑，将短信平台参数协同告警信息封装为 form 数据向短信平台的 URL 发送 POST 请求，短信平台作为中转站接收到请求后会将报文的数据解析出来并以短信形式发送给对应的手机号。

钉钉通知方式利用钉钉群聊中机器人功能，通过在对应群聊中添加机器人，并将机器人的 URL 地址配置到系统中，从而实现系统后台创建告警时将其封装为 json 报文同步对聊天机器人的 URL 发送 POST 请求，进而通过钉钉服务器对报文的解析中转展示到对应的群聊消息中。

企业微信通知方式利用企业微信群聊机器人功能，群聊中添加的每一个机器人都会生成自己的 URL 地址，将机器人的 URL 地址配置到系统中，当系统环境发生告警时，后台会同步将告警消息封装为 json 报文对聊天机器人的 URL 发送 POST 请求，进而通过企业微信服务器对报文的解析中转展示到对应的群聊消息中。

Microsoft Teams 通知方式与钉钉类似，它是利用团队频道中的连接器功能，通过在对应团队频道中添加连接器，并将连接器的 URL 地址配置到本产品中，从而实现系统后台创建告警时将其封装为 json 报文同步对连接器的 URL 发送 POST 请求，进而通过 Teams 服务器对报文的解析中转展示到对应的频道帖子中。

SNMP 是一套由管理员系统监管被管理系统各类上报消息的网络协议，在告警通知的应用场景中，本系统作为被管理的设备环境，除了记录 SNMP 的账

号密码等认证信息之外，还会根据环境告警数据的结构特点开发设计对应的 MIB 文件（信息管理数据库），并将其下载导入到接收告警通知消息的管理员系统 server 端，该文件内部将不同类型的告警消息的请求上报简化映射为对应的 OID 码，server 对接收到来自本系统上报的 OID 码，根据预设的 MIB 文件进行解析翻译，最终转换为意义明确的告警消息展示在自身平台。

Webhook 通知方式主要应用场景是客户二次开发的系统需要被动（不同于主动调用 SDK 接口查询）接收告警信息。具体的应用过程是，客户在自己的系统中开发专用接口用来处理来自本产品的 webhook 请求报文，并将该接口的 URL 配置到本产品的 webhook 地址列表中，当本产品产生新的告警时会将告警信息封装为特定结构体同步向客户的 URL 发送请求，之后交由客户的接口对请求报文进行解析处理。

需要注意的一个细节是，不论是哪种告警通知方式，如果配置在本产品的 URL 是域名的形式，则需要同步配置 DNS 才可以支持告警消息的正常推送。因此，本产品增加了 URL 解析逻辑，及时提示客户配置 DNS。

5.1.3 性能报表

InCloud Sphere 性能报表以曲线图形式展示了主机和虚拟机 CPU 使用率，内存使用率，磁盘读写速率，网卡速率，存储磁盘读写速率等性能数据，用户可以根据需要查看实时性能数据或者一段时间内的历史性能数据。InCloud Sphere 同时支持性能报表的查询与导出功能，以便进行进一步的分析。

5.1.4 健康巡检

一键系统健康巡检功能是针对系统当前时刻运行状态的检查。检查报告对系统总体运行资源及配置进行分析，结合最佳实践给出配置建议，从而使系统达到最佳配置状态。同时，健康巡检功能可以对系统计算资源、网络资源和存储资源进行分析。计算系统资源整体使用率趋势，出具资源扩容建议，以保障系统稳定

运行。

5.1.5 主机自动发现

主机自动发现基于自定义 SSDP 实现。在添加主机时，管理节点会自动在其所在的网络中进行扫描，并根据主机的状态、系统版本等进行筛选，将所有符合条件的主机的信息都快速地呈现给用户，以方便用户添加。主机自动发现功能简化了主机添加的流程，用户不再需要逐一输入主机的 IP 地址或 IP 段。

5.1.6 扫描闲置虚拟机

InCloud Rail 能够持续记录虚拟机的 CPU 使用率、内存使用率、最近一次关机时间，通过闲置虚拟机检测可以发现 CPU 使用率低于指定值、内存使用率低于指定值、关机状态三者任意满足持续时间大于 15 天、1 个月、两个月、3 个月、6 个月、1 年、两年、3 年的虚拟机清单，这些虚拟机称之为闲置虚拟机。虚拟机长期处于闲置状态，说明用户已经不再需要该虚拟机内运行的业务应用了，建议用户及时删除闲置状态虚拟机，以释放 CPU、内存和占用的存储资源。

5.2 虚拟机迁移

5.2.1 计算迁移

vMotion 是 InCloud Sphere 重要功能，vMotion 提供 VM 灵活性和可用性，可以更改运行虚拟机的计算资源，或者同时更改虚拟机的计算和存储资源。可将正在运行的整个虚拟机从一台服务器迁移到另一台服务器上，且虚拟机保留其网络标识和连接，从而实现无缝迁移。为虚拟机提供了灵活性和可用性，满足业务和最终用户不断增长的需求。

InCloud Sphere 支持异构 CPU Model 的虚拟机迁移。根据虚拟机的指令集可以在兼容的架构下实现虚拟机迁移。

vMotion 优势：

- 在零停机且用户未感知的情况下执行实时迁移

- 不间断自动化优化资源池中的虚拟机
- 在不安排停机时间、不中断业务运营的情况下执行硬件维护
- 主动将虚拟机从故障或运行异常的服务器中移出

vMotion 使用的是预复制迁移（Pre-Copy Migration）,具体原理如下：

1) 系统验证目标服务器的存储器和网络设置是否正确，并保留目标服务器虚拟机的资源；
2) 当虚拟机还在源服务器上运转时，将内存镜像复制到目标服务器上。在这个过程中，InCloud Sphere 会监视内存的变化；
3) 内存复制完成后，大部分的内存镜像已经被复制到目标服务器上，检查内存较复制之前是否发生了变化；
4) 假如发生了变化，InCloud Sphere 会将发生变化的内存重新复制到目标服务器中，并覆盖先前的内存。此时 InCloud Sphere 仍然会监视内存的变化情况；
5) InCloud Sphere 底层持续进行内存复制操作，随着复制次数的增加，所需要复制的数据就会明显减少，复制所消耗的时间就会逐渐变短，内存可能没有足够的时间发生变化。最后当源服务器与目标服务器之间的差异可以忽略不计时，内存复制操作结束；
6) 内存复制完成之后，将 VM 的工作状态切换到目标服务器上，源服务器针对迁移的虚拟机停止工作；将存储从源系统上解锁，并锁定在目标系统上。启动目标服务器，并与存储资源和网络资源链接，同时清除源服务器上的资源。

5.2.2 存储迁移

Storage vMotion 是现有的 VM vMotion 实时迁移功能的一种，即存储的迁

移。实质是虚拟机的磁盘文件迁移到主机可以访问的其他存储上，可以是同资源池或跨资源池的，而不影响 VM 运行。

Storage vMotion 优势：

- 重新分配存储器负载。在零停机的情况下实时迁移磁盘数据，平衡容量，提高存储设备的性能
- 存储维护和重新配置。无需关闭虚拟机，即可将虚拟机数据从设备移出，从而对设备进行维护

Storage vMotion 工作原理：

1) 在目的存储建立与源存储一致的虚拟机磁盘结构
2) 将源存储虚拟磁盘的元数据拷贝到与之对应的目的存储的虚拟磁盘中
3) 当数据拷贝完成后，使用动态数据同步功能，将源存储虚拟磁盘的 active 层数据与目的存储的 active 层数据同步
4) 当源存储与目的存储的 active 层数据同步时，将存储从源系统上解锁，并锁定在目标系统上，使虚拟机的磁盘 I/O 重定向到目的存储的虚拟磁盘上，断开源存储磁盘 I/O
5) 源存储资源进行释放，完成 Storage vMotion

5.3 虚拟机调度

5.3.1 负载均衡

InCloud Sphere 负载均衡是对集群中主机资源的优化管理，对当前集群资源利用进行综合评估，并采取对集群中虚拟机的合理调度，达到对集群中资源的负载均衡，以及在保证性能的前提下通过主机的待机操作（将主机电源关闭）减少能耗。

DRS：提供资源负载均衡机制（默认开启）：

- DRS 提供可配置的 CPU、内存利用阈值，进行定制化的配置；
- DRS 周期性的对集群资源利用情况进行检测评估，并依据评估结果合理调度虚拟机，实现整个集群资源的负载均衡；
- DRS 提供优秀的调度策略，当集群中主机资源利用率超过阈值，DRS 调度算法会对该主机上的虚拟机进行迁移预算评估，并从中选出最佳的虚拟机迁移组合方案，在保证能把主机利用率降到阈值的前提下，选取最小的资源迁移量；
- 可靠的迁移过程，DRS 在迁移虚拟机过程中，会对生成的迁移策略进行排序调度，保证单个主机在同一时刻的迁入/迁出线性执行，防止并发迁移给主机增加过大的迁移压力。

5.3.2 节能调度

DPM（电源管理）：在 DRS 基础上提供节能调度机制：

- DPM 提供可配置的 CPU、内存利用率低阈值；
- DPM 同样会针对集群的资源利用情况，产生待机请求（即：将会把主机电源关闭的请求），或对待机的主机进行电源打开操作；
- 当集群负载较低时，DPM 会向 DRS 提出将利用率低于阈值的主机待机的请求，DRS 收到待机请求，同样会根据集群中主机的负载的情况，进行资源预判评估。从利用率低于阈值的主机中选出合理的一个，在保证剩余主机负载不超过阈值的情况下，将该主机上的虚拟机均衡的迁移到其他主机上，并且成功处理完迁移请求之后，DPM 会在等待一个周期的资源稳定之后，将该主机进行下电待机，节约能耗；
- 当集群负载较高时，且当前集群中的主机不足以满足资源需求时，DRS 会向 DPM 发出打开待机主机电源的请求，DPM 会给出候选的待机主机

列表，DRS 会对给出的待机主机进行资源评估，并选取合适的主机，由 DPM 进行电源打开操作。

5.3.3 动态资源扩展

DRX 动态资源扩展是一种纵向资源扩展机制，系统动态监控已加入 DRX 保护的虚拟机成员的内存、CPU 资源，当监测到资源使用阈值高于设定的阈值时及持续一定时间后，利用虚拟机的可热扩展性以及系统的热添加能力，动态向上调整内存、CPU 资源使其资源使用率下降到阈值以下保证虚拟机中的业务有足够的资源去使用，进而达到业务的持续稳定运行。DRX 是一种持续资源监控及调整机制，持续监控并调整虚拟机实例资源使用。当不再需要这种机制时，将其从 DRX 组保护中移除目标虚拟机实例资源即可。

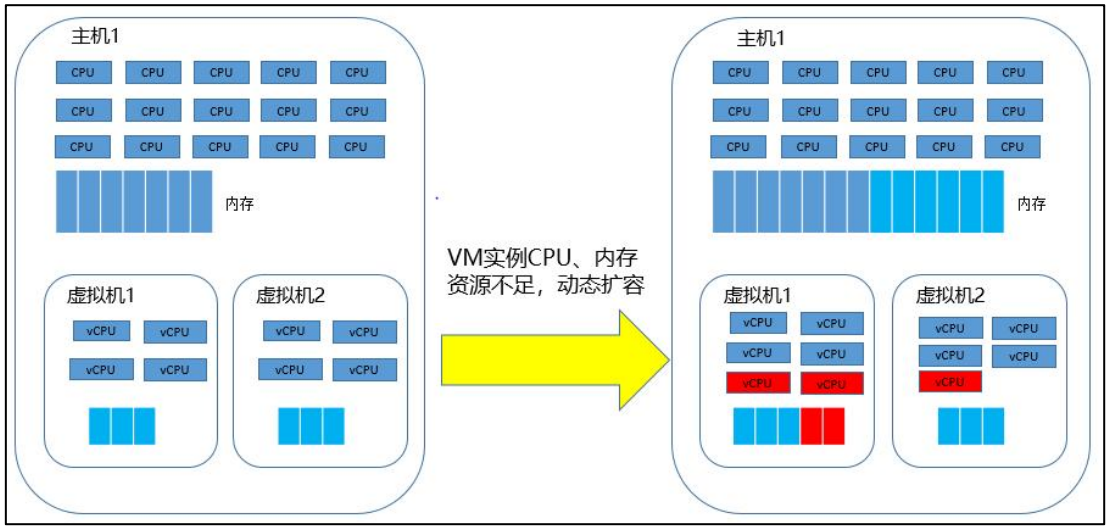


图 5.3.1-1 纵向 DRX

5.4 高可用机制

5.4.1 管理节点高可用

为消除管理节点单点故障，InCloud Sphere 提供管理节点双机热备的部署方式，两台 iCenter 管理节点实时互备数据，其中一台 iCenter 节点作为整个系统的管理节点，当主管理节点宕机时另一台 iCenter 节点会迅速接管整套系统以保

障系统整体的持续运行。

InCloud Sphere 通过 Heartbeat 机制实现了心跳监测和资源接管调度，心跳监测可以通过网络链路和串口进行，而且支持冗余链路，主备管理节点之间相互发送报文以告诉对方自己当前的状态，备管理节点如果在指定的时间内未收到主管理节点发送的报文，则备管理节点就会判定对方失效，进而启动系统接管服务。

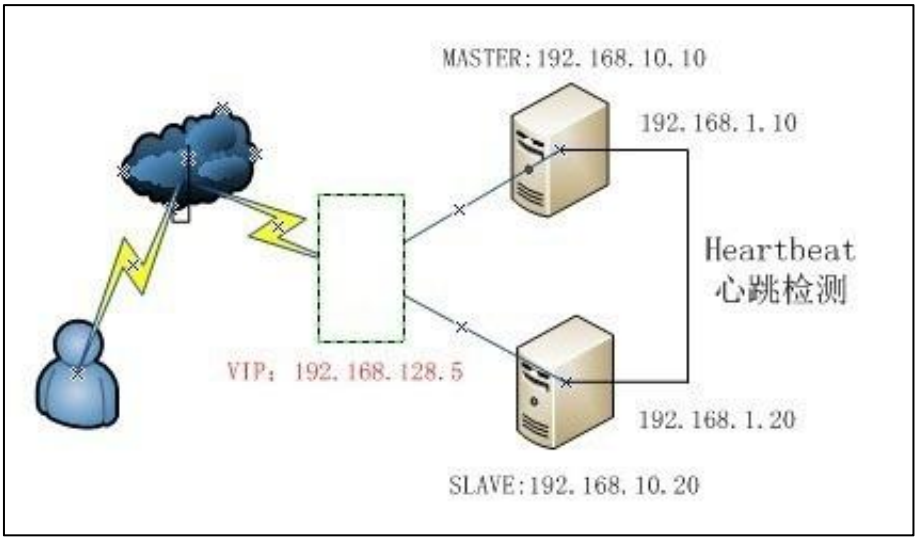


图 5.4.1-1 Heartbeat 技术原理

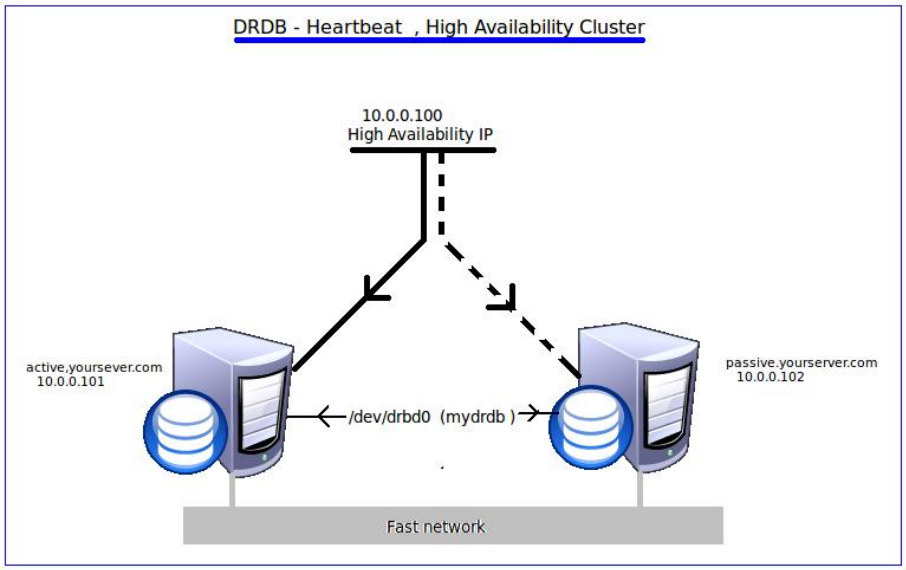


图 5.4.1-2 DRBD 块复制技术原理

InCloud Sphere 基于 DRBD 块设备存储技术主管理节点的数据在更动后自

动复制到备管理节点节点，实现了主备节点间数据块的实时完整复制以达到双机数据的一致性。

Heartbeat 通过读写 DRBD 仲裁盘中的 GI 值以判断主备节点的数据是否一致。DRBD 仲裁盘是一个共享的磁盘，主备节点通过定时服务不停地向该共享盘中写入 DRBD 的 GI 信息。如图 5.4.1-3 所示，GI 信息是 DRBD 数据状态的一种签名，如果主备节点的 DRBD 数据状态是同步的，则 GI 信息值是一样的，反之则不一样。

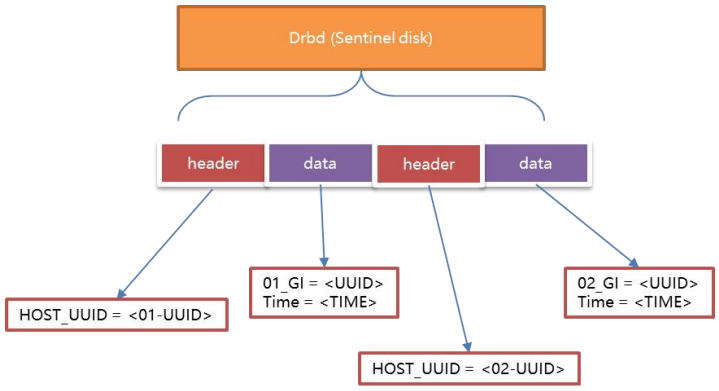


图 5.4.1-3 DRBD 仲裁盘数据结构

当主备管理节点同时意外关机，且其中一台管理节点因硬件故障无法开机时，另一台可以正常开机的管理节点需要通过读取仲裁盘中的 DI 信息以判断其是否可以正常启动管理服务。如果该管理节点的 GI 信息包含对端的 GI 信息，则该节点可以单独启动管理服务，否则禁止启动管理服务。

5.4.2 计算节点升级为主备计算管理节点

服务器出厂预装的节点全部为计算节点，客户现场部署时，可以自动化形式选择两台节点升级部署为主备计算管理节点（AllinOne 节点），从而降低客户现场部署的复杂度。

使用两台已安装 iNode 的服务器节点，选择其中一个节点作为主管理节点，另一个节点作为备管理节点，然后运行升级部署程序。升级部署程序启动后主管

理节点会等待备管理节点连接，两节点建立连接后将同步主机配置信息，然后主备节点各自安装管理节点组件，组件部署完成后两节点分别初始化 AllinOne 配置，最后主节点向备节点同步数据，完成主备节点的建立。

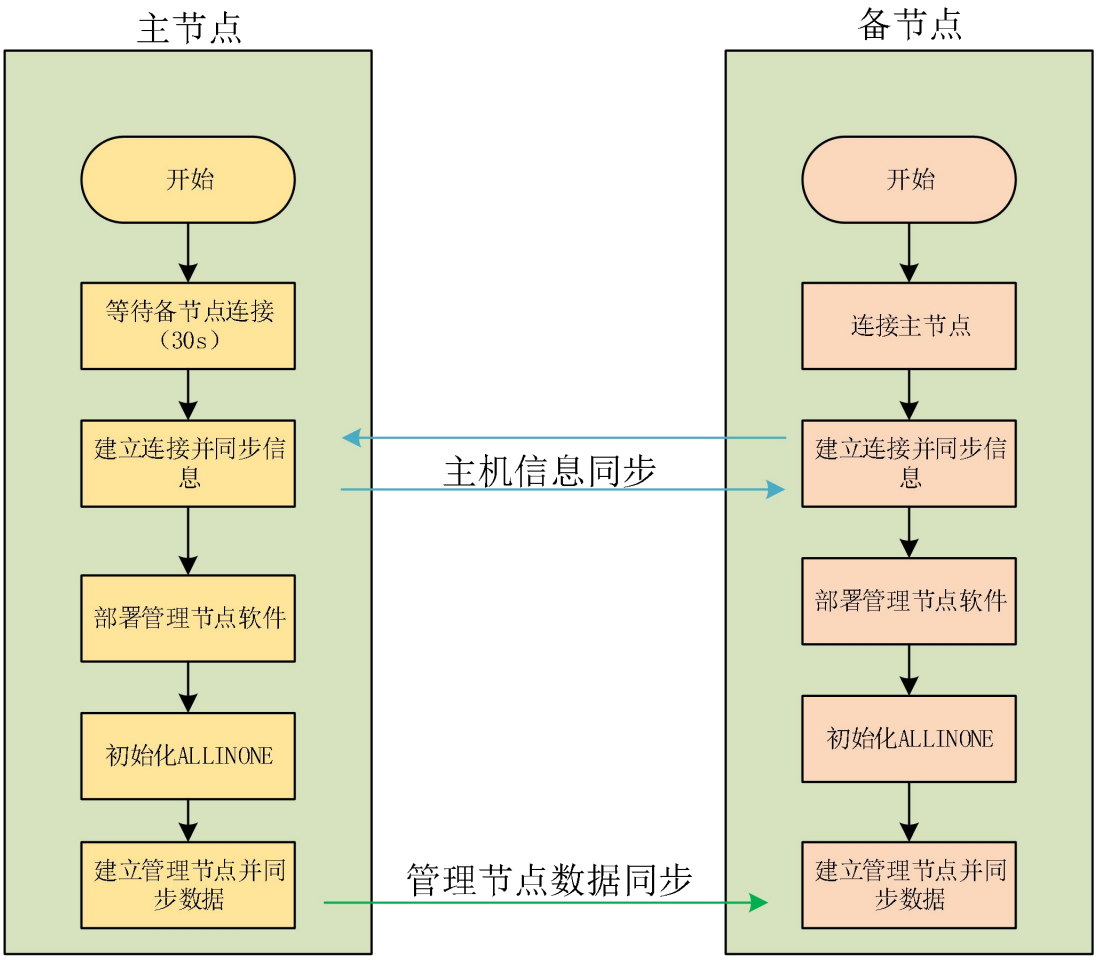


图 5.4.2- 1 主备节点升级部署流程

5.4.3 虚拟机高可用

InCloud Sphere High Availability（InCloud Sphere HA）为系统中虚拟机提供了简单易用、成本效益高的高可用性功能。由于硬件故障导致的 VM 不会再造成灾难性的后果，InCloud Sphere HA 会对宕机的 VM 进行重新分配资源并快速重启这些虚拟机，InCloud Sphere DRS 则会决定放置这些虚拟机的最佳位置以满足其资源要求，其业务可在短时间内恢复正常。

InCloud Sphere 支持虚拟机在异构 CPU 主机间 HA，根据虚拟机的指令集可以在兼容的啊机构的主机上进行 HA。

一、HA 工作原理：

采用 InCloud Sphere HA 主调度服务结合宕机事件主动上报机制实现 InCloud Sphere 的高可用性服务；

InCloud Sphere HA 主调度服务包含 HA 监听器和 HA 事件处理器；HA 监听器用于循环监听 HA 事件处理的结果，若有一次没处理完毕的宕机事件，HA 监听器则继续进行尝试处理；HA 事件处理器用于接收并处理主动上报的宕机事件；

InCloud Sphere HA 宕机事件主动上报机制，服务器或虚拟机发生宕机事件，并将事件快速主动上报给 HA 主调度服务器的 HA 事件处理器。

二、HA 接入控制原理：

集群 HA 接入控制策略提供两种选项，一是 HA 资源预留（默认选项），二是故障切换主机。

其中 HA 资源预留策略通过静态定义 CPU 或内存资源预留百分比实现。当集群中 CPU 或者内存资源剩余百分比小于所配置的 CPU 或内存资源预留百分比，将拒绝新的虚拟机启动或者其他集群的虚拟机在线迁移到该集群，当集群中发生主机宕机或者虚拟机宕机事件，则宕机的虚拟机可以利用为 HA 所预留的 CPU 或内存资源重新启动。

故障切换主机策略通过静态定义故障切换主机列表实现。当集群中部分主机发生故障。HA 将尝试在任一指定的故障切换主机上重新启动其虚拟机。为了确保故障切换主机上拥有可用的空闲容量，手动迁移虚拟机到故障切换主机的功能将禁止。可以手动迁移虚拟机到其他非故障虚拟机。

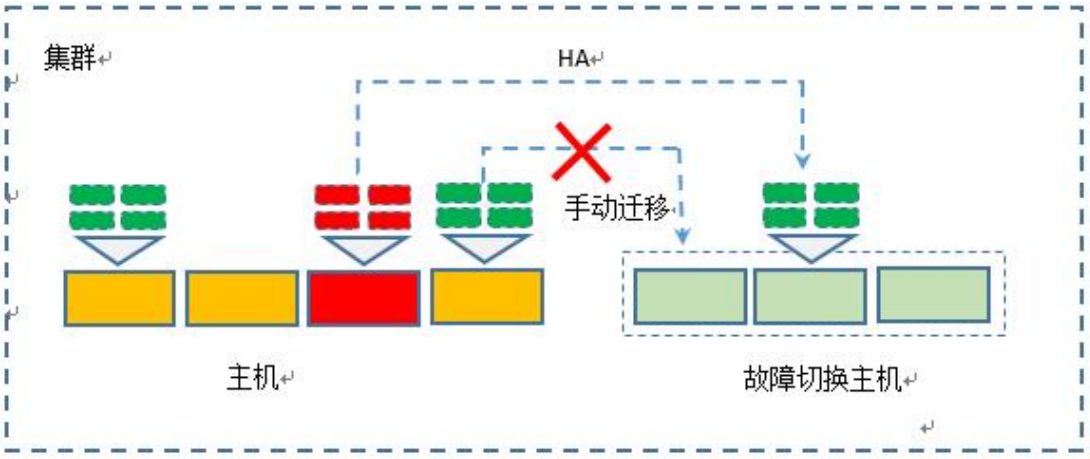


图 5.4.3-1 接入控制主机策略

具体实现方式：

通过 iCenter 配置启用 HA 策略，InCloud Sphere 提供两种设置方式：

（1）iCenter 中集群管理中设置集群 HA 策略；

（2）单虚拟机设置启用 HA;

集群内服务器发生宕机，通过主动上报机制将宕机事件上报，HA 主调度服务将该服务器上运行状态的虚拟机在集群内其它服务器上重新分配资源并进行重新启动。

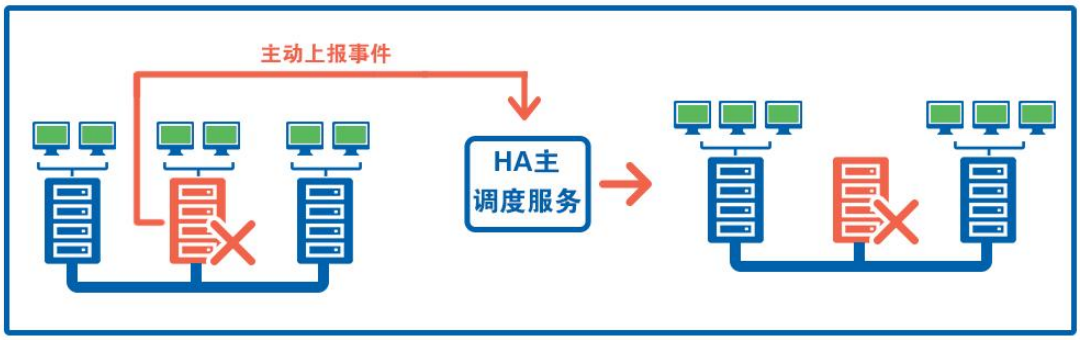


图 5.4.3-2 主机 HA

单个运行状态的虚拟机发生宕机，通过主动上报机制将宕机事件上报，HA 主调度服务对宕机的虚拟机快速进行重启，如图 5.4.3-3。

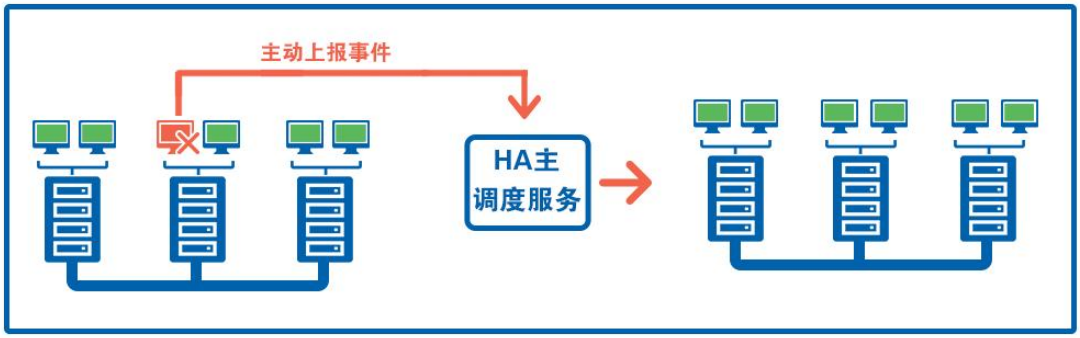


图 5.4.3-3 虚拟机 HA

5.4.4 带裸 LUN 的虚拟机高可用

InCloud Sphere 支持挂载裸 LUN（FC、iSCSI）的虚拟机在所运行的主机异常关机时， 能够在其他同集群的并且添加了此存储适配器的主机上自动启动此虚拟机，保证虚拟机业务的高可用。

带裸 LUN 的虚拟机 HA 需满足以下前置条件：1）虚拟机打开迁移许可；2）虚拟机系统盘在共享存储上。

5.4.5 虚拟机崩溃恢复策略

InCloud Sphere 对虚拟机发生蓝屏或宕机时支持多种处理策略方式。当部分 Windows 系列操作系统的虚拟机出现蓝屏和部分 Linux 系列操作系统的虚拟机出现宕机，InCloud Sphere 能够执行不同的恢复策略。策略生效的前提是虚拟机需要安装 VM Tools。

不处理：即虚拟机出现蓝屏或宕机，不对这个虚拟机做任何处理；

重启：即虚拟机出现蓝屏或宕机，InCloud Sphere 将自动重启虚拟机；

关机：即虚拟机出现蓝屏或宕机，InCloud Sphere 将自动关闭虚拟机；

HA：即虚拟机出现蓝屏或宕机，InCloud Sphere 将自动拉起虚拟机。

5.4.6 面向业务网的虚拟机 HA

InCloud Sphere 实时监测计算节点的网卡状态，当网卡发生故障断开导致虚

拟机业务网络不可用时，InCloud Sphere 自动将该业务网络关联的虚拟机迁移至其他主机上运行，从而减少因为主机网卡断开造成的业务中断时间。

InCloud Sphere 结合底层上报的机制，在计算节点网卡故障上报时将故障信息记录在数据库，HA 处理模块通过解析故障信息，过滤出该计算节点发生业务网断开的虚拟机，然后给这些虚拟机匹配出目标主机，并进一步根据用户配置的业务网 HA 容忍度及处理策略，来触发 HA 任务将其迁移至目标主机上运行。

5.5 内置灾备机制

虚拟机备份功能旨在允许用户从虚拟机备份恢复由于服务器硬件故障损坏或用户误操作损坏的虚拟机。

5.5.1 虚拟机备份

管理员通过虚拟机备份功能，为虚拟机在可用存储上创建虚拟机某一时刻的备份。特定类型的虚拟机不支持虚拟机备份： 包含裸磁盘的虚拟机，包含物理网卡或 SRIOV 网卡的虚拟机。支持为启动和关闭状态的虚拟机创建备份。

InCloud Sphere 使用了独有专利的基于差分的备份方式，不依赖于虚拟机快照，备份不会影响虚拟机的性能。

支持备份的存储类型有本地存储、CFS 存储和 NFS 存储。挂载厚置备磁盘的虚拟机不支持备份到 NFS 存储。由于本地存储位于特定主机，所以不推荐使用本地存储作为备份存储，推荐使用 CFS 存储或 NFS 存储（如果虚拟机没有厚置备磁盘）。

系统支持设置自动为虚拟机创建备份功能，通过此功能，系统可以在指定时间为指定的虚拟机创建备份。可以设置只创建一次或周期性执行。设置调度任务后，系统会在设置的时间自动为指定的虚拟机创建备份，自动备份任务的成功与否依赖备份时的主机状态和存储状态，如果主机，网络和存储状态良好，则可以成功生成虚拟机备份。支持虚拟机批量备份。

用户使用备份恢复功能，用指定的虚拟机备份恢复生成新虚拟机。恢复备份生成的虚拟机与源虚拟机配置相同，虚拟磁盘内的数据状态为源虚拟机创建备份时的文件系统状态。

恢复备份产生的虚拟机是一个全新的虚拟机（拥有全新的虚拟机名称和 ID），恢复备份过程不依赖源虚拟机状态，生成的新虚拟机与源虚拟机互不影响。

5.5.2 CBT 备份技术

CBT 备份基于数据块变化追踪（Changed Block Tracking）技术，为虚拟机增量备份提供了基础，除第一次备份必须完整备份外，后续的增量备份都只需通过查询 CBT 位图，只需备份变化过的区块，节省了区块数据差异对比时间，所需传输的数据量也大幅减少。

CBT 备份方式的特点是数据最全面、最完整，数据一致性得到完全保护，备份的效率高，备份时间短。如果灾难发生时，可以恢复到任意备份点，不再受限于其他备份文件，每个备份点之间是相互独立。

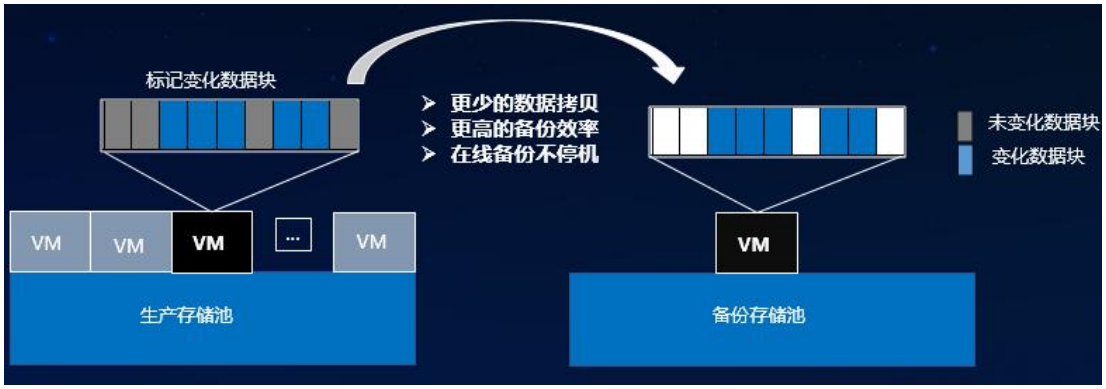


图 5.5.2-1 CBT 备份

5.5.3 备份计划任务

InCloud Sphere 在备份的基础上实现了自动化的备份调度任务。通过设置计划任务，实现备份工作的自动化调度进行，用户通过自定义备份时间点，选择适合的备份时机。对于关键业务虚拟机进一步增加了业务的安全性。

5.5.4 持续数据保护

持续数据保护（Continuous Data Protection，简称 CDP）是一种在不影响主要数据运行的前提下，可以实现持续捕捉或跟踪目标数据所发生的任何改变，并且能够恢复到此前任意时间点的方法。

InCloud Sphere 通过植入后端存储过滤驱动，来实时捕获虚拟磁盘所有 IO 访问操作。基于写时复制（Copy on Write，简称 CoW）机制，在虚拟机真实 IO 写入磁盘之前，将变化数据拷贝至备份存储池中。当进行数据恢复时，虚拟机录影机依据 IO 时间戳，从备份存储池把备份的 IO 回写到当前虚拟机磁盘中。具体实现如下图所示：

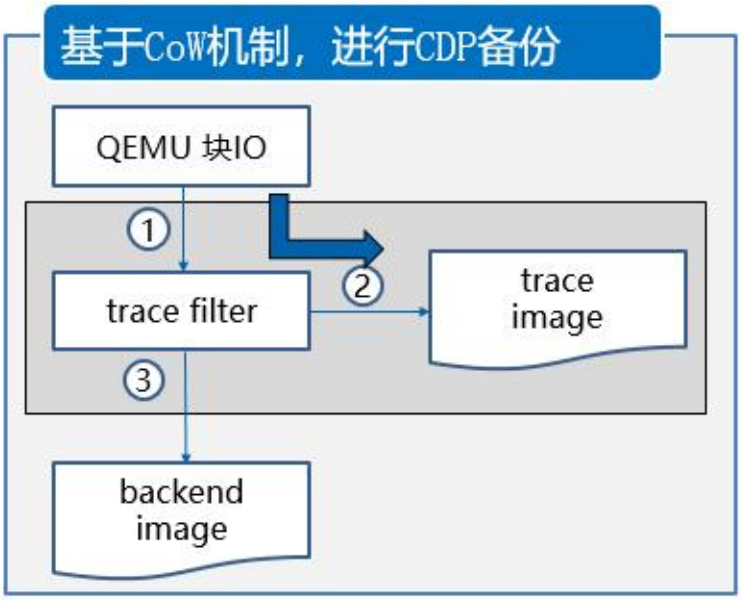


图 5.5.4-1 持续数据保护原理

① 存储过滤驱动截获写入源磁盘时的 IO；

② 新 IO 记录连同当前时间戳自动保存到备份存储池中；

③ 新 IO 记录写入虚拟机源磁盘。

5.5.5 存储双活技术

提高系统稳定性和应用灵活性，保证关键业务的连续性是虚拟化技术的一个

重要特性。InCloud Sphere+智能存储 G2/G5 的双活数据中心解决方案即使如此。

双活数据中心需要在数据中心从上到下各个层面都要实现双活，即存储、服务器、网络、数据库、应用各层面都具有双活设计，这样才是在真正意义上实现数据中心层面的双活。

InCloud Sphere+智能存储 G2/G5 的双活数据中心相结合的模式，打造出独特的双活数据中心解决方案。方案利用 G2/G5 跨数据中心的存储虚拟化功能和数据镜像功能，结合上层 ICS 应用集群，使两个数据中心都处于运行状态，可同时承担相同业务，从而提高数据中心的整体服务能力和系统资源利用率。同时使用独立的仲裁服务器作为防止应用系统中存储脑裂的发生，依赖第三方的仲裁点实现对应用中存储故障的处理，当单数据中心故障时，业务自动切换到另一数据中心，实现 RPO=0，RTO≈0，解决了传统灾备中心不能承载业务和业务无法自动切换的问题。

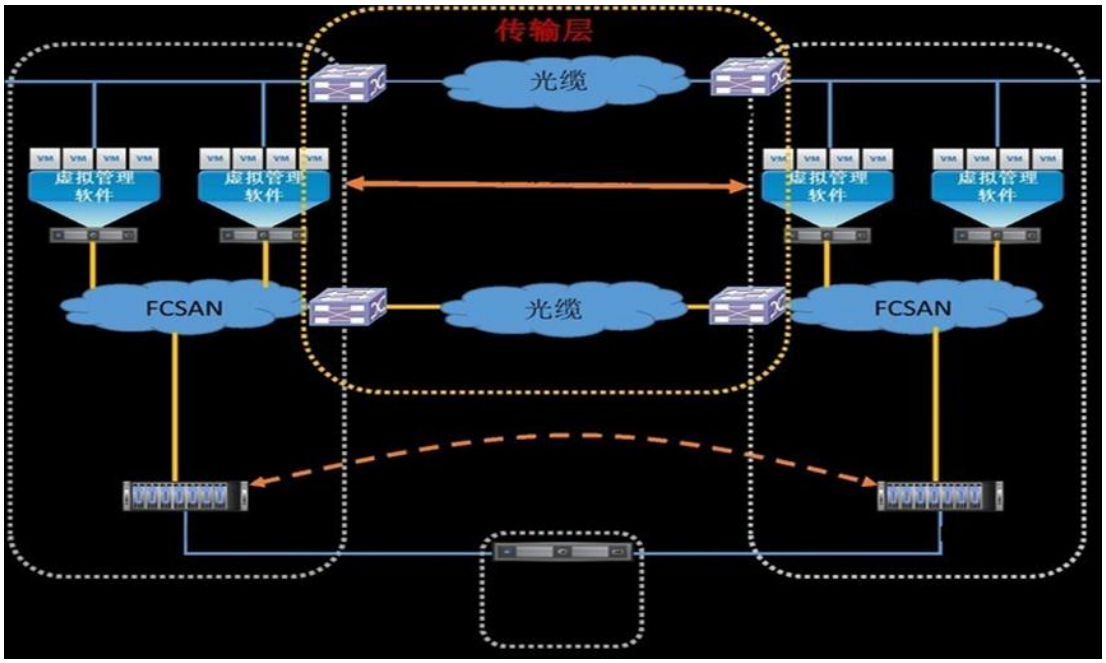


图 5.5.5-1 InCloud Sphere+智能存储 G2/G5 双活

5.5.6 虚拟机数据容灾

虚拟机数据容灾是指因不可逆的自然或者硬件故障等意外原因，导致原集群

中所有运行虚拟机的硬件不可用，将存储数据或备份数据尽快恢复至新的生产环境中，使生产得以继续。对应到 InCloud Sphere 虚拟化解决方案中可以理解为如何从存储中恢复虚拟机数据，进而保证生产业务连续性/恢复。

InCloud Sphere 支持数据容灾功能，可以保证意外故障下虚拟机数据及其业务的快速上线。

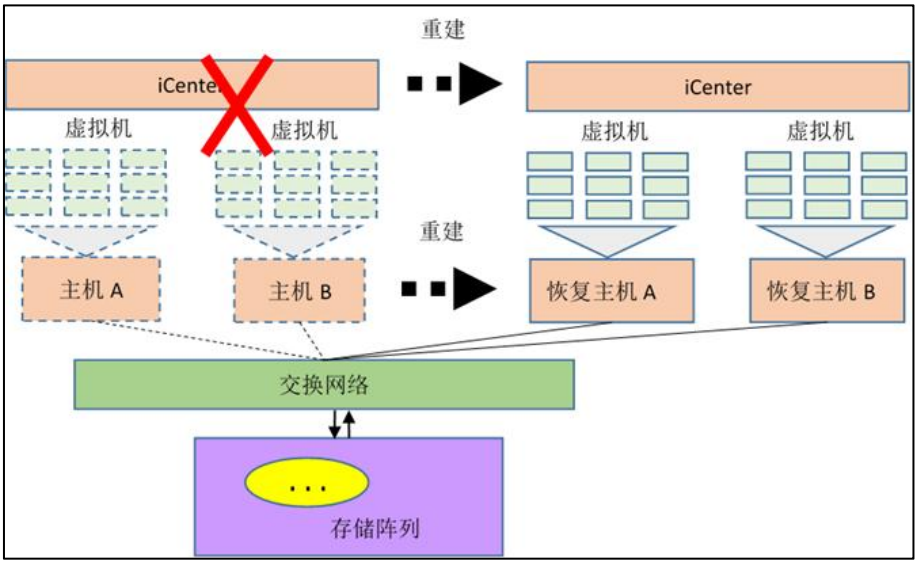


图 5.5.6-1 虚拟机数据容灾

故障主机与恢复主机访问共享存储，虚拟机虚拟磁盘使用共享存储。故障发生前虚拟机生命周期管理期间，共享存储管理保存虚拟机元数据信息并周期同步。一旦 iCenter、主机整体故障，在恢复主机上同步注册并管理映射的共享存储数据。恢复主机扫描共享存储数据上的虚拟机数据信息，并注册虚拟机到新的主机上。至此在故障恢复主机完成虚拟机管理及业务上线。

InCloud Sphere 虚拟机数据容灾技术，结合存储双活技术能够提供共享存储级别的容灾功能，满足客户对生产环境下的多种容灾场景的不同需求。

5.5.7 系统备份与备份恢复

InCloud Sphere 企业版提供了系统备份与恢复功能，备份支持每天系统自动备份与手动立即备份功能。系统自动备份是 iCenter 管理程序在每天的固定时间

将 iCenter 等的管理元数据自动备份，自动备份策略最长保存 7 天的系统备份数据。系统自动备份的备份目录支持 iCenter 本地目录、iNode 目录与第三方 FTP 服务器，可以自定义更改备份目录，默认备份目录为/var/lib/backup/。立即备份可以备份当前时刻的系统管理元数据。立即备份会循环覆盖最近一次的备份，仅保留一份立即备份产生的备份数据。备份数据仅支持备份到系统本地。

备份恢复可以将 iCenter 管理元数据恢复到上次系统备份文件生成时刻的状态，保障 iCenter 管理元数据异常时能够快速恢复客户管理数据。恢复时可以使用 iCenter 本地或用户本地保存的系统备份完成备份恢复。

5.5.8 双站点容灾

InCloud Sphere 支持双站点容灾功能（SRM），包括主站点和备用站点，在主站点发生故障时，备用站点可以接管服务，实现系统的持续稳定运行。将位于两地的 InCloud Sphere 站点设置互为主备，站点容灾保护下的虚拟机的虚拟磁盘自动在主备站点之间同步，当虚拟机运行的站点发生故障后，可以将虚拟机业务切换到异地站点运行。

双站点容灾功能（SRM）功能通过 CDP（持续数据保护）技术实现，实现原理为虚拟机在执行写 IO 时，系统自动捕获虚拟机的 IO 信息，将写 IO 信息封装为 IO 日志缓存文件，InCloud Sphere 系统自动将 IO 日志缓存文件传输到异地站点， 异地站点将 IO 日志合入到本地的虚拟磁盘中，从而实现异地站点之间的数据同步。可通过 RPO（Recovery Point Objective）配置项设置站点之间的同步频率，系统自动将一个 RPO 周期内的虚拟机 IO 封装为一个 IO 日志缓存文件发送到异地站点。

配置双站点容灾功能(SRM)需在两个站点的每一台主机配置缓存存储池，被保护的虚拟机的 IO 日志缓存文件被暂时保存在缓存存储池中。站点之间需要打通三层网络（容灾网络），两个站点上的计算节点可以通过容灾网络互通，通

过容灾网络直接进行增量数据的传输。

5.6 系统安全机制

5.6.1 虚拟机安全隔离

通过 Hypervisor 严格的资源隔离机制，不同的虚拟机具有独立的 CPU、内存、磁盘和网络资源。分配给一台虚拟机的内存和磁盘中的数据不会被其他虚拟机所读取。虚拟机释放内存后，内存中的数据在被严格清空后才会被其他虚拟机使用。虚拟机删除虚拟磁盘后，磁盘中的数据会被严格清空以防止被主机系统或其他虚拟机读取。

5.6.2 基于角色的访问控制

InCloud Sphere 支持基于角色的访问控制，角色细粒度地定义可以访问的资源类型，及允许对该资源类型执行的操作。例如是否能对主机执行增加、删除、密码认证、挂载存储、进入维护模式等。通过将角色赋予用户，用户便具备了对资源的操作权限。同时，可以将用户归类到用户组中，统一为用户组中所有的用户分配角色、配置权限。

5.6.3 用户操作审计

InCloud Sphere 操作日志能够记录用户在系统中的所有操作，包括操作的名称、详细描述、执行操作的用户名称、操作对象、操作结果、执行操作的开始时间和结束时间，及对失败操作的原因分析。同时，只有被赋予权限的用户才可能访问操作日志。

5.6.4 系统登录控制

InCloud Sphere 通过设置多种策略，保证用户对系统的合法登录。通过密码策略，设置允许的密码的有效期、长度，以及特殊字符、大小写、数字和字母的位数；通过锁定策略，设置登录尝试失败的最多次数和解锁时间等；通过会话策略，设置可以并发访问系统的用户数量，及会话保持的最长时间等。

5.6.5 虚拟机管理员角色

虚拟机管理员角色（VMManager）是面向虚拟机及其相关资源等的权限管理机制。针对虚拟机、虚拟机模板、虚拟机备份、虚拟机任务等资源的相关操作权限进行管理。

虚拟机管理员角色（VMManager）与 Administrator、Readonly 角色一样是系统初始化之后的默认角色，不可更改与删除。虚拟机管理员角色（VMManager）与 Administrator、Readonly 等角色互斥，当用户具有上述角色权限或者其他自建角色权限时，需要手动解除 Administrator 或者自建角色后才能授予用户虚拟机管理员角色（Readonly 角色自动从用户移除，无需手动移除）。同样当用户已经具有虚拟机管理员角色，此时授予其他角色时需要手动解除虚拟机管理员角色后，才能授予其他角色权限。当用户被授予虚拟机管理员角色权限后，用户需要重新登录后才能操作属于该用户的虚拟机及其相关资源。虚拟机具有“所属用户”属性，创建虚拟机后，所属用户默认为创建虚拟机时的登陆用户。当用户操作不属于该用户本身的虚拟机时，系统会提示“正在操作其他用户的资源，请确认是否继续执行”的提示。

5.6.6 第三方 LDAP 认证

InCloud Sphere 第三方认证支持无缝接入 LDAP 认证协议，第三方 LDAP 域中的用户以手动、自动以及周期性自动的方式同步到系统用户列表中，无需在系统中重复创建或维护域用户信息，降低用户信息运维成本。

在系统中增加 LDAP 认证源时需要验证连通性，若连通则在系统中配置对接第三方 LDAP 域，建立用户关系映射表，自动同步域用户到系统用户列表中。第三方 LDAP 域中新增或删除用户时，支持手动以及周期性自动更新系统用户列表，保持系统中的域用户信息与第三方 LDAP 域中一致。系统中的域用户默认授予只读角色，为域用户授权相关角色后，可以使用域用户直接登录系统进行操作使用。针对 LDAP 域服务器信息变化，支持修改注册到系统的配置信息，

如服务器地址、是否 SSL/TLS 加密、服务器端口号、管理员 DN 和管理员密码等，保证系统到第三方 LDAP 域的连通性。删除 LDAP 域后，自动清空系统用户列表中的域用户，并清除域用户与其它资源的关联信息，不能再通过这些用户登录到系统。

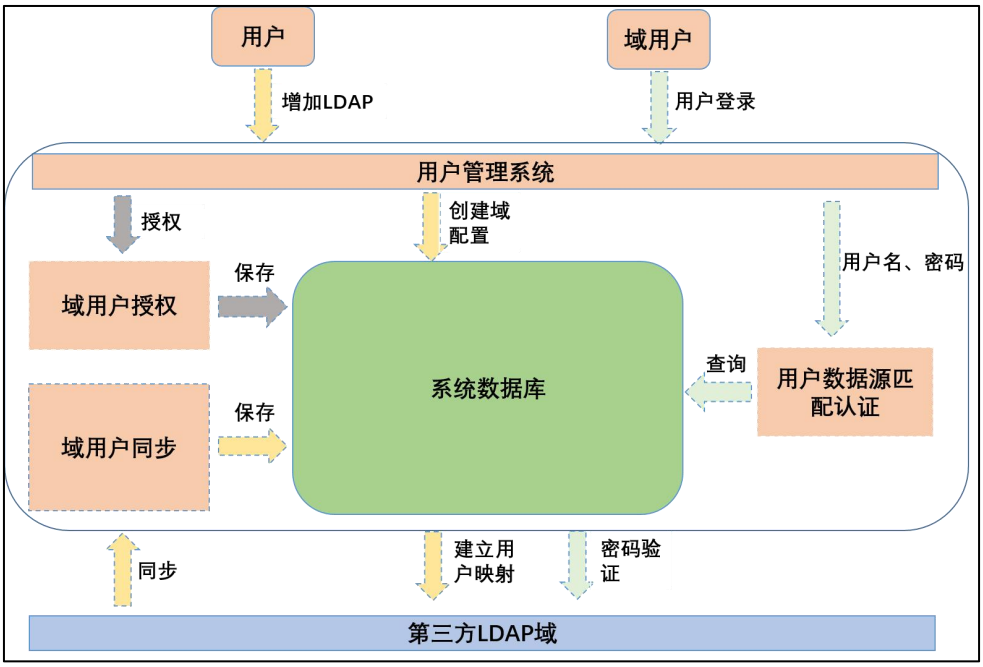


图 5.6.6-1 第三方 LDAP 认证原理

5.6.7 AK/SK 认证

AK/SK 是一种身份认证访问方式，其中 AK 为 Access Key ID，SK 为 Secret Access Key，通过 AK/SK，用户无需密码即可直接访问系统的 API。AK/SK 密钥对与系统中的用户角色绑定，不同的 AK/SK 密钥对具有不同的资源访问与操作权限。

每个用户可以生成多个 AK/SK，可以随时启用、禁用或者删除自己创建的 AK/SK，AK/SK 具有该用户完全的权限。生成的 AK/SK 密钥对包括：Access Key ID、Secret Access Key、启用状态、所有者和生成时间。后续可以直接使用 AK/SK 访问系统的 API 接口，而无需用户密码认证，使用方法为在发送请求的

Herders 表单中传入 Key 值 Authorization，对应的 Value 格式为(ICS {Access Key ID}:{Secret Access Key})，另外可以通过 SDK 的方式配置 AK/SK 访问系统。在 AK/SK 认证时通过其中的 Access Key ID 来标识用户，使用 Secret Access Key 密钥来验证用户，该密钥必须保密，不同用户只能生成、查看、操作属于自己的 AK/SK 密钥对。AK/SK 是对 API 请求进行安全验证的关键因子，需要妥善保管，若某一 AK/SK 出现泄漏风险，建议及时禁用或删除该 AK/SK 并生成新的 AK/SK。

5.6.8 OTP 认证

一次性密码（One Time Password，简称 OTP），OTP 认证作为一次性密码认证，用于账户密码登录后的第二次认证，提高系统安全性。OTP 基于 TOTP（Time-based One-Time Password）算法每隔一定的时间间隔生成一个不可预测的随机数字组合。OTP 的密码有效期仅在一次会话中（通常为 30 秒），因此不容易受到重放攻击。OTP 一般分为计次使用和计时使用两种，采用计时使用的 OTP 则可设置密码有效时间，从 30 秒到两分钟不等，而 OTP 在进行认证之后即废弃不用，降低了不经授权访问限制资源可能性。

OTP 认证逻辑是在服务端为每位用户生成一个共享密钥，作为生成一次性密码的关键值。用户通过扫描二维码或手动添加在 APP 终端记录该共享密钥。APP 终端通过 TOTP 算法通过共享密钥和当前时间计算出一次性密码。服务端也根据相同的算法和共享密钥以及当前时间，生成一个密码与客户端密码进行比较，如果匹配成功则验证通过。所以 OTP 认证对服务端与 APP 终端上的时间保持一致有着很高的要求（误差不超过 30 秒）。常见的 TOTP 应用包括 Google Authenticator 和 Authenticator 等身份验证应用程序，它们生成基于 TOTP 算法的动态密码用于进行一次性密码验证。

5.7 升级管理

InCloud Sphere 提供界面化的升级管理功能，为 iCenter 和 iNode 提供安全

便捷的升级操作指导。

5.7.1 更新包管理

为实现新特性的扩展，及对一些问题的修正。浪潮为 InCloud Sphere 提供可靠的更新包与更新包管理机制。可以从配置的下载源下载更新包或通过界面上传更新包两种方式对系统内的管理节点与计算节点升级。更新包分为：升级包与补丁包。升级包是版本的升级，升级后系统的版本会发生变化；补丁包是补丁的升级，升级后，系统版本不变。每个更新包都带有包版本、包类型、应用产品、发布日期等信息。

5.7.2 安全便捷升级

InCloud Sphere 系统管理员，可在发布升级包之后将升级包下载到本地，并通过安全方式上传到 iCenter 中对系统的各个元素进行升级；升级操作会优先进行升级包校验、升级主机状态检测等系列列安全检查，确保升级过程的顺利进行；另外 iCenter 升级操作的一致性，保证升级元素所有组件的完整升级，而无需担心升级部分组件导致的系统不一致，为系统升级的安全一致提供了可靠保证。升级过程可选关闭虚拟机或者可选迁移虚拟机对生产业务进行管理。

5.7.3 升级日志维护

InCloud Sphere 自动记录每一次升级信息，保留每一次升级日志，方便用户来查看系统的升级历史。同时支持导出升级日志，方便系统维护。

5.8 多云管理

多云管理可以纳管 VMware 和其他版本 InCloud Sphere 的虚拟机的管理，通过在多云管理中添加要管理的站点，实现对该站点中的虚拟机进行管理。

5.8.1 VMware 虚拟机管理

InCloud Sphere 集成了 VMware 虚拟机管理模块，通过添加 vSphere 站点，用户在 InCloud Sphere 管理界面即可实现对 VMware 虚拟机的监管，查看虚拟

机的电源状态、操作系统，并实时监控虚拟机的 CPU 和内存利用率。

5.8.2 VMware 虚拟机迁移

InCloud Sphere 实现了虚拟机从 vSphere 环境到 KVM 环境的在线迁移，用户无需担忧迁移过程业务中断，亦无需购买额外的第三方迁移服务。

VMware 虚拟机在线迁移至 InCloud Sphere 技术原理：在 InCloud Sphere 上创建一个新空白磁盘，将 VMware 虚拟机通过快照复制的方式，不断把虚拟机数据同步到 InCloud Sphere 平台，当脏数据（指 VMware 平台虚拟机内存和 InCloud Sphere 平台虚拟机内存中不相同的数据）足够小时，关闭 VMware 虚拟机，并将最后一次增量数据同步到 InCloud Sphere 平台，最后在 InCloud Sphere 平台开启虚拟机。

在线迁移可划分为以下步骤：

1、InCloud Sphere 获取待迁移虚拟机的配置信息；

2、将 VMware 虚拟机的虚拟磁盘迁移到 InCloud Sphere 存储池；

3、VMware 虚拟机持续创建内存快照和磁盘快照，并将快照文件迁移到 InCloud Sphere 存储池；

4、迁移过程中 InCloud Sphere 记录 VMware 虚拟机由于数据写入产生的内存和磁盘脏数据；

5、当脏数据少于指定阈值时，暂停 VMware 虚拟机，剩余脏数据全部复制到 InCloud Sphere 环境；

6、关闭 VMware 虚拟机，启动 KVM 虚拟机，完成用户业务切换。

5.8.3 虚拟机跨云迁移

浪潮 InCloud Sphere 通过多云管理管理模块为用户提供跨多个云平台进行

各类云资源统一管控的功能,包含 VMware 虚拟机迁移至 InCloud Sphere 平台,InCloud Sphere 系统虚拟机在不同的 InCloud Sphere 平台间互相迁移。

InCloud Sphere 跨云迁移虚拟机,打破平台界限,使得不同平台版本,不同地域的数据中心资源的维护与更加便利。虚拟机迁移支持在线迁移和离线迁移,其中虚拟机在线迁移过程中业务不会中断,亦无需购买额外的第三方迁移服务。

InCloud Sphere 跨云迁移虚拟机,分为在线迁移和离线迁移两类。在线迁移依赖 libvirt daemon 服务,实现计算资源和存储资源的同时迁移。离线迁移是基于底层打开的数据传输通道来同步存储数据。迁移过程中同步进行信息汇总,并展示迁移进度,同时支持迁移任务的撤销。

InCloud Sphere 虚拟机在线迁移是通过 libvirt client 发起,源主机与目标主机端 libvirt daemon 协同完成。迁移过程无需人工干预,同时完成计算资源与存储资源的迁移。在线迁移不支持虚拟机带快照迁移。

InCloud Sphere 虚拟机离线迁移主要是存储资源的迁移,在源主机与目标主机端打通数据传输通道,按顺序拷贝磁盘数据,传输过程中数据经过加密。存储数据远程拷贝完毕后,在目标主机重建并启动虚拟机。

多云管理 iCenter 虚拟机迁移方案灵活扩展平台的资源管理,支持在线迁移和离线迁移。迁移任务可以随时撤销,并且实时展示迁移进度。

适用于对业务的承载资源调整,或者业务负载均衡等需要虚拟机迁移的场景。

使用约束:

1、InCloud Sphere 要求 5.8.1 或者之后的更新版本。

2、在实际业务环境中,由于网络带宽及虚拟机文件大小等不同,在线迁移可能触发超时机制;业务允许的情况下,可以转为离线迁移。

3、在线迁移不支持虚拟机带快照迁移。

4、启动虚拟机在线迁流程之前，停止待迁移虚拟机周期性任务。

5.8.4 虚拟机分发到边缘站点

虚拟机从中心站点主机迁移至边缘站点主机，支持在线迁移及离线迁移。

InCloud Sphere 虚拟机在线分发是通过 libvirt client 发起，源主机与目标主机端 libvirt daemon 协同完成，分发过程无需人工干预，同时完成计算资源与存储资源的迁移。

InCloud Sphere 虚拟机离线分发主要是存储资源的迁移，在源主机与目标主机端打通数据传输通道，按顺序拷贝磁盘数据，传输过程中数据经过加密。存储数据远程拷贝完毕后，在目标主机重建并启动虚拟机。

5.8.5 I2V 迁移

ICS 虚拟机迁移到 VMware 平台是浪潮云海服务器虚拟化系统 InCloud Sphere 上多云管理的一个功能。多厂商、异构云平台的统一管控是多云管理的基础，浪潮云海服务器虚拟化系统 InCloud Sphere 可为用户提供跨多个云平台，各类云资源统一管理的功能。

InCloud Sphere 实现了虚拟机从 InCloud Sphere 环境到 VMware vSphere 环境的离线或在线迁移，用户无需购买额外的第三方迁移服务。

InCloud Sphere 虚拟机离线迁移至 VMware 技术原理：在 VMware 上创建一个占位虚拟机，将 InCloud Sphere 虚拟机通过 VMware 的 vddk 和 InCloud Sphere 的 vvdk，把虚拟机数据同步到 VMware 平台，最后在 VMware 平台开启虚拟机。

InCloud Sphere 虚拟机在线迁移至 VMware 技术原理：在 VMware 上创建一个占位虚拟机，将 InCloud Sphere 虚拟机通过快照复制的方式基于 VMware 的 vddk 和 InCloud Sphere 的 vvdk，把虚拟机数据不断同步到 VMware 平台，

当脏数据(脏数据是指 InCloud Sphere 平台虚拟机运行过程中不断修改、但是还未复制到 VMware 平台虚拟机的磁盘数据)足够小时，暂停 InCloud Sphere 虚拟机，并将最后一次增量数据同步到 VMware 平台，最后在 InCloud Sphere 平台关闭虚拟机，在 VMware 平台开启虚拟机。

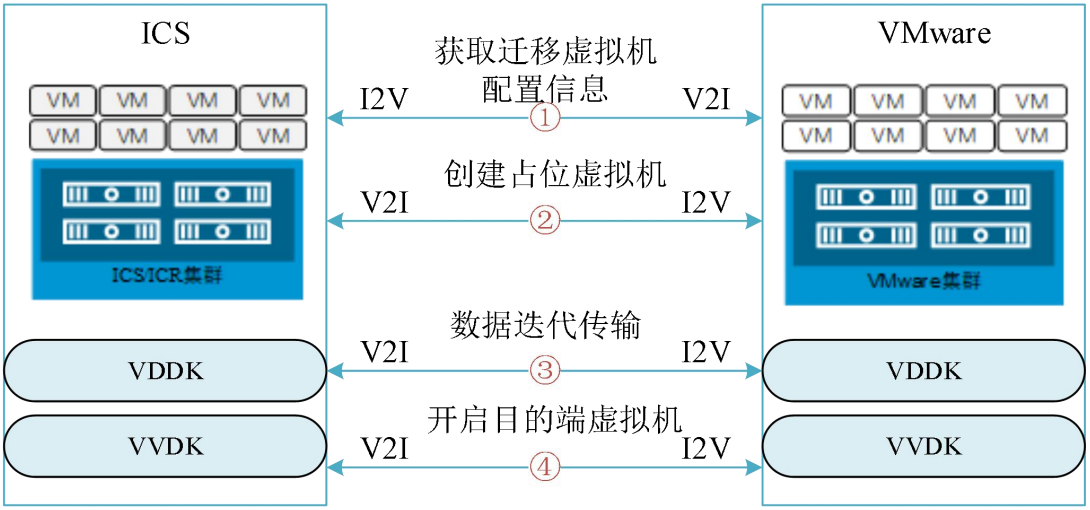


图 5.8.5-1 I2V 迁移原理

离线迁移的原理可划分为以下步骤：

1. 关闭 InCloud Sphere 平台虚拟机；
2. InCloud Sphere 获取待迁移虚拟机的配置信息；
3. 基于获取的配置信息在目的端创建站位虚拟机接收迁移数据；
4. 将运行于 InCloud Sphere 平台上的虚拟机的虚拟磁盘复制到 VMware 存储池的空白磁盘；
5. 在 VMware 平台开启虚拟机，完成 InCloud Sphere 平台虚拟机到 VMware 平台的迁移。

在线迁移的原理可划分为以下步骤：

1. InCloud Sphere 获取待迁移虚拟机的配置信息，在 VMware 创建同样配

置的虚拟机；

2. 将运行于 InCloud Sphere 平台上的虚拟机的虚拟磁盘复制到 VMware 占位虚拟机对应的空白磁盘；

3. InCloud Sphere 平台虚拟机持续创建磁盘快照，并调用 VMware 的 vddk 将快照数据迁移到 VMware 存储池，迁移过程中 InCloud Sphere 记录迁移虚拟机由于数据写入产生的磁盘脏数据；

4. 当脏数据小于指定阈值时，暂停 InCloud Sphere 平台虚拟机，剩余脏数据全部复制到 VMware 环境；

5. 关闭 InCloud Sphere 平台虚拟机；

6. 在 VMware 平台开启虚拟机，完成用户业务切换。

6 北向开放接口

InCloud Sphere 提供全功能的 API，为第三方云平台厂商以及 OpenStack 提供接口，具有为云计算提供 Hypervisor 的能力，针对不同用户对云平台的需求提供接口，使得用户可以采用浪潮云平台产品以及第三方的云平台产品对自己的业务统一管理

6.1 REST API

6.1.1 REST API 介绍

REST API 是 InCloud Sphere iCenter 对外提供管理接口的统称，是 InCloud Sphere 虚拟化管理的核心，由一系列的 HTTP 请求组成。

REST API 主要提供 iCenter 端 Manager 服务的客户端查询、操作 API。客户端应用包括 WEB 浏览器用户接口，通过 REST API 来获取 InCloud Sphere iCenter 端信息；并通过 REST API 来管理 InCloud Sphere iCenter 服

务。客户端应用根据业务需要调用 iCenter 服务的相应业务 REST API，ICS Manager 服务接收 HTTP 请求并根据请求做出正确响应，完成相应业务。例如：用户通过用户接口添加 InCloud Sphere 虚拟化管理服务 License，用户通过主流浏览器操作 InCloud Sphere Web Client 界面（用户接口）；Web Client（用户接口）调用添加 License 的 REST API，发起 HTTP 请求（REST 请求）；ICS Manager 服务接收 HTTP 请求，并处理请求，验证 License，为系统添加 License ；产生响应反馈给 Web Client 端；Web Client 根据响应数据转换为用户可读信息。

简而言之，REST API 就是客户端和 InCloud Sphere iCenter 端 ICS Manager 服务通信的接口。

6.1.2 REST API 功能

REST API 功能如下：

- REST API 主要提供客户端与 ICS Manager 服务通信的接口；
- REST API 是有一组 HTTP 请求组成，便于各种语言开发的客户端接入操作；
- REST API 允许用户使用任何支持 HTTP 请求的客户端来与 iCenter 端 ICS Manager 服务交换；
- iCenter 端 ICS Manager 服务对外仅提供 REST API 接口，便于管理和控制对外暴露接口，增强系统安全，易于维护；
- REST API 作为 InCloud Sphere 产品对外暴露功能接口，方便上层合作方了解我们产品提供功能，便于根据需要进行功能扩展。

6.1.3 REST API 架构

REST API 是基于 HTTP 协议，使用标准的 HTTP 方法，比如 GET、PUT、

POST 和 DELETE，进行通信。REST 是指一组架构约束和原则，客户端和服务器之间的交互在请求之间是无状态的。REST API 架构 WEB 服务，每个资源都有一个地址。资源本身都是标准 HTTP 方便调用操作的目标。可以使用各种语言（比如 Java 程序、Perl、Ruby、Python、PHP 和 JavaScript）实现客户端。各种语言编写的可以通过 REST API 与 InCloud Sphere iCenter 通信。

6.2 Java SDK

Java SDK 为 InCloud Sphere 提供的管理功能接口，通过 SDK 调用，发送 REST API 请求与 iCenter 进行交互，模拟人工界面操作。通过集成 Java SDK，可实现对 InCloud Sphere 的功能逻辑控制，方便将虚拟化管理无缝集成至其他管理平台。InCloud Sphere Java SDK 具有以下优势：

1) 接口简单：开发者无需了解交互时采用的协议、加密、校验等问题，只需了解接口功能以及参数即可完成对应的功能。

2) 功能模块清晰：SDK 采用分模块方式构建，相近操作类型归纳至统一模块管理，层次清晰，便于开发者了解业务模型，降低了学习成本。

3) 接口丰富：SDK 接口覆盖 iCenter 全部功能点，无需对数据进行额外复杂处理，即可完成功能调用。

4) 无缝迁移生产环境：集成 SDK 于测试环境中开发调试完成后，可让开发人员调试完成后无修改直接接入生产环境。

7 应用场景

7.1 基础设施集中化

InCloud Sphere 通过基础设施集中化，为单一机构、远程或分支机构提供多种类型的虚拟操作系统和应用程序；基于虚拟化技术，提高硬件整合度和资源共享度，提升基础设施整体效率；为虚拟化操作系统和应用程序提供了强可用性、可恢复性和安全性。

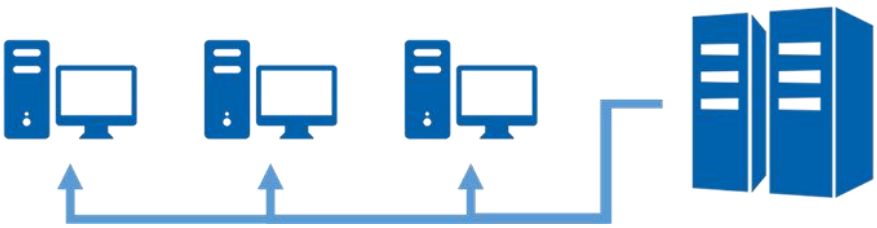


图 7.1-1 基础设施集中化

通过基础设施集中化，InCloud Sphere 架构为用户解决了以下痛点：

1. 机房空间不足

InCloud Sphere 架构支持 1:20 的服务器与 VM 的整合比，降低设备数量，解决机房空间不足问题。

2. 资源利用率低下

InCloud Sphere 架构基于一虚多特性，利用率从 10%提升到 70%，将业务应用整合集中，大幅提高服务器 CPU、内存利用率。

3. 老旧系统不兼容

对无法在旧服务器上继续维护老旧业务系统，InCloud Sphere 可通过 P2V 将业务转换迁移至虚拟化平台，保证业务继续稳定运行。

4. 数据中心整合

InCloud Sphere 将多个数据中心的业务统一整合到一个数据中心，实现资源的物理集中化。

7.2 基础设施现代化

基础设施现代化亦可称之为私有云，以标准的形式提供数据中心资源的 IT 效率，以自动化提升资源利用的灵活性和效率。

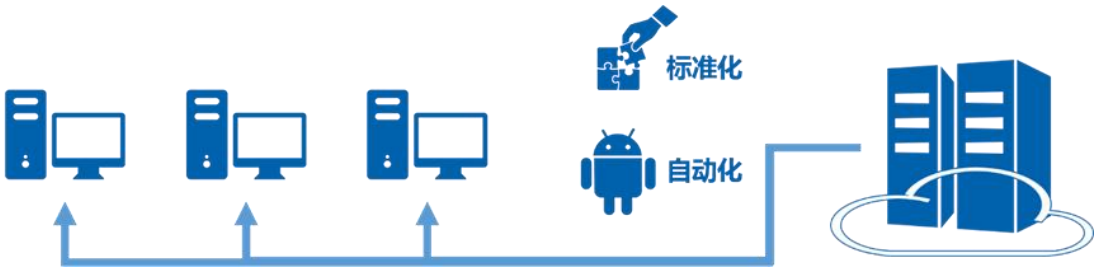


图 7.2-1 基础设施现代化

通过基础设施现代化，InCloud Sphere 为用户提供了以下收益：

1. 降低机房能耗

InCloud Sphere 的 DPM 和 DRS 功能相互配合，自动实现业务集中化，关闭资源利用率低的机器以降低能耗。

2. 保障业务连续性

InCloud Sphere 通过 HA 将意外关机的 VM 自动重启；通过 DRS 保障业务能有充足的资源可用，进而保障业务稳定运行。

3. 保障数据安全

InCloud Sphere 通过快照将重要数据保存在虚拟化系统内的存储设备上；备份将重要数据保存到虚拟化系统之外，进一步提高数据安全性。

4. 测试开发环境

InCloud Sphere 将测试基准环境制作成虚拟机模板或镜像，从而极大降低测试基础工作量，提高效率。

7.3 基础设施敏捷化

基础设施敏捷化强调资源供给的敏捷性，面向敏捷开发、按需快速扩展，以满足云原生应用、业务创新和用户需求的多样性与云提供商的交互能力。

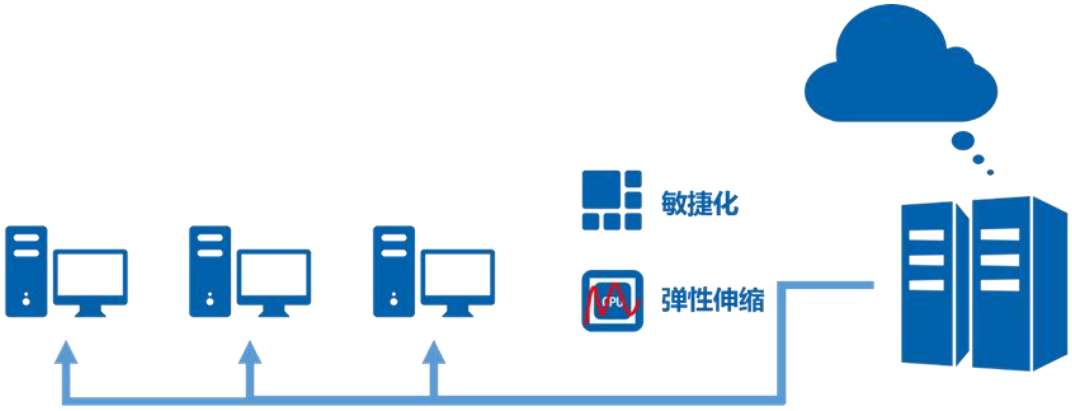


图 7.3-1 基础设施敏捷化

通过基础设施敏捷化，InCloud Sphere 为用户提供了以下收益：

1. 提升业务灵活性

InCloud Sphere 帮助用户摒弃业务与服务器的绑定关系，降低业务应用和硬件的耦合度，基于资源池实现资源供给，极大提升业务的灵活性。

2. 新业务快速上线

InCloud Sphere 帮助用户摒弃传统的业务上线再购买搭建新的服务器资源的做法，将要批量部署的业务制作成模板或镜像，实现业务快速部署。

3. 业务负载快速响应

InCloud Sphere 帮助用户能够实时监控 CPU 和内存利用率，基于资源需求和业务优先级可快速生成资源分配策略。

8 总结

浪潮秉承以客户为关注焦点，技术创新是原动力理念，关注客户需求，通过技术创新和产品创新，为客户提供完善的产品和方案，InCloud Sphere 即是浪潮众多产品的核心一员。通过本文档我们了解到通用虚拟化技术的原理，InCloud Sphere 的功能和架构，InCloud Sphere 在资源虚拟化、自动化运维和开放性三个方面的技术特性。InCloud Sphere 是浪潮为客户倾力打造的战略级产品，为客户提供应用向虚拟化平台的部署或迁移，构建云数据中心的关键能力。

关于更多 InCloud Sphere 的资料和内容，请联系相关行业和区域浪潮人员。

9 术语表

表 9- 1 术语表

英文缩写	英文全称	中文全称
API	Application Programming Interface	应用软件编程接口
CPU	Central Processing Unit	中央处理器
DHCP	Dynamic Host Configuration Protocol	动态主机配置协议
DNS	Domain Name System	域名系统
DRS	Distributed Resource Scheduler	分布式资源调度
VS	Virtual Switch	虚拟交换机
EPT	Extended Page Table	扩展页表
FC	Fiber Channel	光纤通道
FCoE	Fibre Channel over Ethernet	以太网光纤通道
FC SAN	Fiber Channel Storage Area Network	光纤通道存储区域网络
FTP	File Transfer Protocol	文件传输协议
GPA	Guest Physical Address	客户机物理地址
GPU	Graphics Processing Unit	图形处理器
GVA	Guest Virtual Address	客户机虚拟地址
HA	High Availability	高可靠性
HBA	Host Bus Adapter	主机总线适配器
HPA	Host Physical Address	主机物理地址
HTTP	Hypertext Transfer Protocol	超文本传输协议

英文缩写	英文全称	中文全称
HVM	Hardware Virtual Machine	硬件虚拟机
IaaS	Infrastructure as a Service	基础设施即服务
IDC	Internet Data Center	互联网数据中心
I/O	Input and Output	输入输出
IOPS	Input/Output Operations Per Second	每秒读写（I/O）操作次数
IP	Internet Protocol	网络互连协议
IPMI	Intelligent Platform Management Interface	智能平台管理接口
IP SAN	IP Storage Area Network IP	存储区域网络
IPSec	Internet Protocol Security	互联网传输协议安全性
iSCSI	Internet Small Computer Systems Interface	因特网小型计算机系统接口
ISV	Independent Software Vendor	独立软件供应商
JDK	Java Development Kit	Java 开发工具包
JMS	Java Message Service	Java 消息服务
KVM	Kernel-based Virtual Machine	基于内核的虚拟机
LACP	Link Aggregation Control Protocol	链路汇聚控制协议
LAN	Local Area Network	局域网
LDAP	Lightweight Directory Access Protocol	轻量目录访问协议
LUN	Logical Unit Number	逻辑单元编号
LVM	Logical Volume Manager	逻辑卷管理器
MAC	Media Access Control	介质访问控制层

英文缩写	英文全称	中文全称
MAN	Metropolitan Area Network	城域网
MMU	Memory Management Unit	内存管理单元
MTU	Maximum Transmission Unit	最大传输单位
NAPT	Network Address Port Translation	网络端口地址转换
NAS	Network Attached Storage	网络附属存储
NAT	Network Address Translation	网络地址转换
NFS	Network File System	网络文件系统
NIC	Network Interface Controller	网卡
NTP	Network Time Protocol	网络时间协议
OS	Operating System	操作系统
OVA	Open Virtualization Appliance	开放虚拟化设备
OVF	Open Virtualization Format	开放虚拟化格式
PaaS	Platform as a Service	平台即服务
PXE	Preboot Execute Environment	预启动执行环境
RAID	Redundant Array of Independent Disks	独立冗余磁盘阵列
QoS	Quality of Service	服务质量
RAM	Random Access Memory	内存
RBAC	Role-Based Access Control	基于角色的访问控制
RDP	Remote Desktop Protocol	远程桌面协议
REST	Representational State Transfer	表示状态转移
RPO	Recovery Point Objective	数据恢复点

英文缩写	英文全称	中文全称
RTO	Recovery Time Objective	业务恢复时间
PV	Paravirtualization	半虚拟化
SaaS	Software as a Service	软件即服务
SAN	Storage Area Network	存储区域网络
SCSI	Small Computer Systems Interface	小型计算机系统接口
SMB	Server Message Block	服务器信息块
SNAT	Source Network Address Translation	源地址转换
SOAP	Simple Object Access Protocol	简单对象访问协议
SSL	Secure Sockets Layer	安全套接层
TCP	Transmission Control Protocol	传输控制协议
UDP	User Datagram Protocol	用户数据报协议
UEFI	Unified Extensible Firmware Interface	统一的可扩展固件接口
UI	User Interface	用户界面
URL	Uniform Resource Locator	统一资源定位器
USB	Universal Serial Bus	通用串行总线
VDC	Virtual Data Center	虚拟数据中心
VDI	Virtual Desktop Infrastructure	虚拟桌面基础结构
VEB	Virtual Ethernet Bridge	虚拟以太网桥
VEPA	Virtual Ethernet Port Aggregator	虚拟以太网端口聚合
VIF	Virtual Interface	虚拟接口

英文缩写	英文全称	中文全称
VLAN	Virtual Local Area Network	虚拟局域网
VM	Virtual Machine	虚拟机
VMM	Virtual Machine Monitor	虚拟机监控器
VPC	Virtual Private Cloud	虚拟私有云
VPN	Virtual Private Network	虚拟私有网络
VSP	Virtual Switch Port	虚拟交换机端口
vSS	Standard Virtual Switch	标准虚拟交换机
vSwitch	Virtual Switch	虚拟交换机
WAN	Wide Area Network	广域网
WOL	Wake On Lan	LAN 唤醒功能
XML	Extensible Markup Language	扩展标记语言