InCloud Sphere 4.5 旗舰版

技术白皮书 V1.0

浪潮（北京）电子信息产品有限公司

2017 年 1 月

目　录

1 第一章 摘要

浪潮，着力推动中国“行业云”，致力于成为中国领先的云计算解决方案供应商，业已形成涵盖 IaaS、PaaS、SaaS 三个层面的整体解决方案服务能力。

浪潮凭借高端服务器、海量存储、云操作系统、信息安全技术为客户打造领先的云基础 架构，基于浪潮企业、行业、政务信息化软件、终端产品和解决方案，全面支撑政务云、行 业云、企业云建设，被评为“中国云计算创新典范企业”。

浪潮 InCloud Sphere 虚拟化产品是云计算基础平台的核心组成部分，通过服务器虚拟化技术将存储、网络和其他外设有机地结合到一起，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，除了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，主要为企业核心业务向云计算迁移、构建企业云数据中心提供了虚拟化技术和能力。

该技术白皮书首先介绍了浪潮 InCloud Sphere 虚拟化产品；其次对浪潮 InCloud Sphere 旗舰版的系统设计、核心技术进行说明，重点对 CPU 虚拟化、内存虚拟化、I/O 设备虚拟化三方面进行描述；然后对浪潮 InCloud Sphere 旗舰版的功能原理进行了详细说明，在计算、存储、网络、高可用、负载均衡等方面的高级功能特性，分别描述了其功能场景、技术原理 等内容；最后突出介绍了 InCloud Sphere 虚拟化产品在自动化能力、开放性和安全性上具有的技术领先优势，展现了 InCloud Sphere 虚拟化平台在服务器整合、互联网数据中心（IDC）和桌面云（VDI）三大解决方案中提供的核心技术能力，及虚拟化技术给用户带来的使用价值。

2 第二章 InCloud Sphere 产品概述

通过阅读本章，您可以了解到：

Ÿ InCloud Sphere 旗舰版产品在客户 IT 环境的定位和作用；

Ÿ InCloud Sphere 旗舰版产品的构成；

2.1 InCloud Sphere 介 绍

InCloud Sphere 旗舰版是浪潮推出的一套企业级开放式服务器虚拟化解决平台。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，从而大大降低数据中心成本。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 旗舰版同时还是一套完整的虚拟化基础架构解决方案。随着云计算和虚拟化技术向构建新一代数据中心方向发展，实现了以虚拟化为基础，兼有实时迁移功能的 64 位系统管理程序、功能全面的管理控制台，数据中心所需的各种运维管理工具和开放性 API。满足企业关键应用向虚拟化平台迁移对于资源高性能、高可靠、高安全和扩展性上的要求，优化基础架构，提高自动化管理水平，进一步减少企业 IT 的整体投入。

InCloud Sphere 旗舰版基于Xen 设计，本身就是一种具有高可靠性、高可用性和高安全性的虚拟化平台，同时还提供了精简部署，管理节点不单独占用一个物理机或虚拟机，并且针对 Windows 和 Linux 虚拟服务器进行了优化，能够提供与本地应用不相上下的性能和无与伦比的虚拟机密度。InCloud Sphere 旗舰版虚拟化平台重点聚焦企业核心业务的需求，提供满足 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的虚拟化技术，加快推动业务和应用的云化。

客户使用 InCloud Sphere 旗舰版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 4.5 旗舰版的主要特性包括：

Ø 高性能

基于Xen 设计，Xen 可以在一套物理硬件上安全的执行多个虚拟机，它和操作平台结合的极为密切，以高性能、占用资源少著称；针对 Windows 和 Linux 虚拟服务器操作系统

进行了优化，提供不亚于裸机的性能；虚拟机的 CPU、内存和 I/O 设备等通过 CPU 绑定、EPT、GPU 直通、WLB 等技术进一步增强虚拟机性能。

Ø 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，降低硬件(内存)采购成本。在存储虚拟化上，采用存储的快速克隆、存储瘦分配技术，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，

降低硬件(存储)采购成本。

Ø 高可用

InCloud Sphere 4.5 旗舰版的 vMotion 和 Storage vMotion 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 4.5 旗舰版均可自动重启服务器承载虚拟机，保护所有的虚拟化应用。支持虚拟 CPU、虚拟内存、虚拟磁盘、虚拟网卡的热插拔功能，减少系统计划内宕机时间，并为企业带来更高的可用性。

Ø 管理控制台

同时提供 C/S、B/S 两种架构的 InCloud Sphere iCenter 管理控制台，方便企业灵活选择部署；两种架构的 iCenter 分别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 Windows 桌面部署 C/S 架构的管理控制台，或者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。

Ø 安全性

可以同时承载多个不同 OS 系统的虚拟机，并实现数据处理、网络连接和数据存储的安全隔离；支持虚拟化无代理杀毒解决方案，有效的解决了虚拟化有代理杀毒模式的安全风暴 问题，从最底层保障了虚拟化系统的信息安全，大大降低了安全系统对虚拟平台的影响，保

证虚拟化系统的高效利用和可靠性；资源池内的所有连接升级使用 TLS 1.2 安全协议，进一步提高了数据安全性。

Ø 扩展性

提供完整的 XAPI 和软件开发工具包 SDK，完全兼容 OpenStack 架构，并可以根据不同客户需求进行 XAPI 的扩展和开发；提供 Introspect API，自由扩展使用第三方安全产品能够监视虚拟基础结构并保护其免受恶意活动攻击；提供与微软Systems Center VMM和Systems Center Opration Manager实现集成管理的能力。

2.2 InCloud Sphere 架构

InCloud Sphere 4.5 旗舰版的体系架构如图 2.2-1 所示：

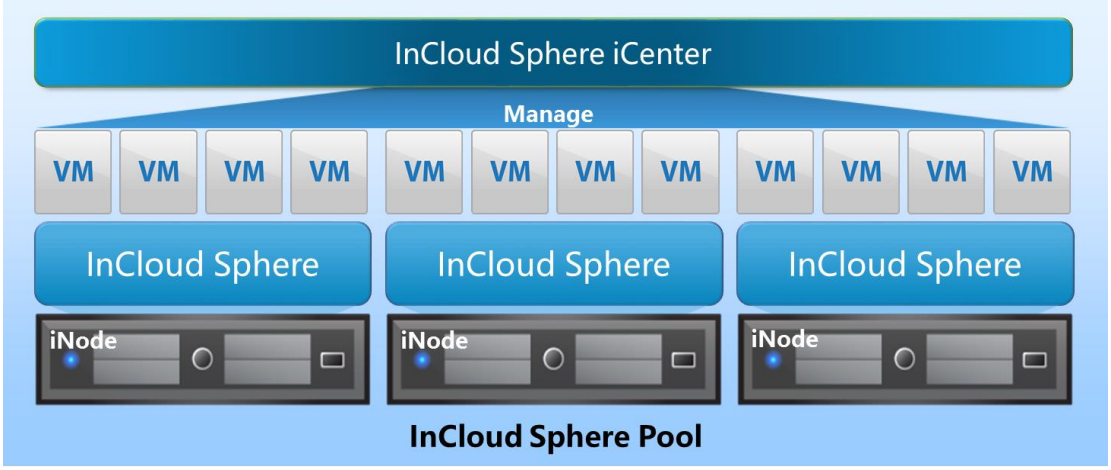


图 2.2-1 InCloud Sphere 4.5 旗舰版逻辑架构

对图 2.2-1 InCloud Sphere 4.5 旗舰版逻辑架构中的不同组件详细介绍如下：

Ø InCloud Sphere iCenter 是为系统管理员提供的一款通用的系统管理工具，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。目前 iCenter 支持 B/S 和 C/S 两种架构，方便企业灵活选择部署。iCenter 可以同时连接多个服务器和资源池，多个 iCenter 也可以同时操作同一个资源池。

Ø VM 是 Virtual Machine 的简称，即虚拟机，有时候也被称为来宾虚拟机（Guest VM）。运行在 InCloud Sphere 4.5 旗舰版上的完整计算机系统，可以提供与传统物理机相同的功能和性能。

Ø InCloud Sphere 是 InCloud Sphere 4.5 旗舰版实现虚拟化的核心，直接安装在物理硬件之上，将硬件层抽象虚拟化，同时控制虚拟机的执行。安装了 InCloud Sphere 4.5 旗舰版的物理服务器被称为 InCloud Sphere iNode，它提供环境所需的硬件资源，包含 CPU、内存、存储和网络等。

Ø InCloud Sphere Pool 多个 InCloud Sphere iNode 主机的硬件资源及其承载的虚拟资源在一起被称为 InCloud Sphere Pool（资源池），资源池内的所有 iNode 主机采用同样的网络和存储配置，实现了 iNode 主机，存储，虚拟机等资源配置的统一管理；同时还能提供 vMotion、HA 等高级功能。

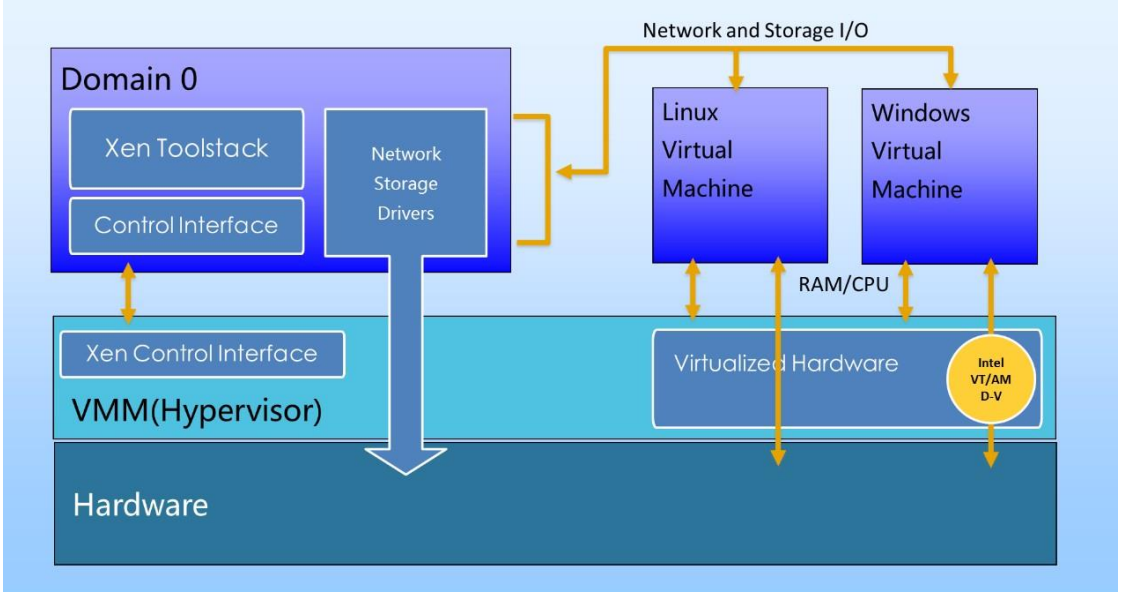
3 第三章 InCloud Sphere 技术原理

通过阅读本章，您可以了解到：

- InCloud Sphere 旗舰版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 InCloud Sphere 系统设计

InCloud Sphere 4.5 旗舰版的体系架构如图 3.1-1 所示：



InCloud Sphere 4.5 旗舰版系统架构

对图 3.1-1 InCloud Sphere 4.5 旗舰版体系架构中的不同组件详细介绍如下：

- **Hardware**为硬件层，包含所有的物理服务器组件（CPU、内存和磁盘驱动器等）
- **VMM**（或称为 Hypervisor）是 Xen 虚拟机管理程序，是运行于硬件上的一个软件薄层。Xen Hypervisor 是一个允许每台物理服务器运行一台或多台“虚拟服务器”的抽象层，有效地将来宾虚拟机及其应用程序与底层硬件分离开来。Xen Hypervisor 不仅抽象出硬件层，同时控制虚拟机的执行。
- **Domain 0**（或称为 Control Domain）是一个运行在 Hypervisor 上的特权 Linux 虚拟机。对硬件而言，具有比其他来宾操作系统更高的优先级。Domain 0 管理所有来宾虚拟机的网络和存储 I/O，而且由于它使用的是 Linux 设备驱动程序，所以能广泛支持各种物

理设备。因为 Domain 0 需要和其他的虚拟机进行交互，因此需要在其它来宾虚拟机启动之前启动

- **Xen Toolstack** 是 InCloud Sphere 4.5 旗舰版中的一个工具栈（toolstack），能够让用户完成虚拟机的创建、删除、配置等功能。此工具栈还提供了一个访问接口，因此，其管理 功能还可以通过相应的命令行工具、图形化控制台或者如 Cloudstack 或 Openstack 类的云计算环境来完成。一系列的 toolstack 组成了 XAPI，为 InCloud Sphere 4.5 旗舰版的核心，提供了 iCenter 和资源池中各主机通信的接口。iCenter 通过 XAPI 来进行 InCloud Sphere 4.5 旗舰版的配置、管理、数据库的维护等，同时也包括如存储（SR）、虚拟机、虚拟网卡、HA 等所有的功能的控制；资源池中的操作请求也是通过 XAPI 传递给 dom0， 同时在池中的所有主机之间通信。
- **Linux Virtual Machine** 是系统为 Linux 的来宾虚拟机，为运行在 Xen Hypervisor 上的半虚拟化虚拟机，包括半虚拟化内核和驱动程序。通过 Domain 0 访问存储和网络资源，通过部署在硬件上的 Xen 访问物理 CPU 和内存。
- **Windows Virtual Machine** 是系统为 Windows 的来宾虚拟机，为运行在 Xen Hypervisor 上的全虚拟化虚拟机，使用半虚拟化驱动程序通过 Domain 0 访问存储和网络资源。Xen 在设计上充分利用 Intel-VT 和 AMD-V 硬件虚拟化技术，提高 Windows 内核的性能。

在 InCloud Sphere 4.5 旗舰版中虚拟机与硬件的所有交互都是通过 Domain 0 进行管理的，它本身就是一个在系统管理程序上运行的具有特别权限的虚拟机。InCloud Sphere 4.5 旗舰版的 Domain 0 如图 3.1-1 左上所示，Domain 0 运行经过优化的 Linux 实例。对管理员而言，Domain 0 是整个 InCloud Sphere 4.5 系统的一部分，不需要额外的安装或管理。Domain 0 使 InCloud Sphere 能够利用标准开源 Linux 设备驱动程序，从而获得了非常广泛的硬件支持。

InCloud Sphere 4.5 旗舰版的系统架构设计中，结合了半虚拟化和硬件辅助虚拟化的优点，允许来宾操作系统在虚拟化硬件上运行。操作系统和虚拟化平台之间的这种协作可帮助 开发者开发更便捷的系统管理程序，同时最大程度地优化系统性能。目前 InCloud Sphere 4.5 旗舰版能够支持多种 Linux 版本的半虚拟化，包括 Red Hat、Novell、SUSE、Debian、Oracle 以及 CentOS。对于不能完全实现半虚拟化的来宾操作系统（例如 Windows 操作系统），InCloud Sphere 可以利用现在 Intel 和 AMD 处理器（Intel-VT 和 AMD-V）中包含的硬件辅助虚拟化技术来实现高性能的虚拟化。

3.2 InCloud Sphere 核心技术

虚拟化（Virtualization）是资源的逻辑表示，而不受物理限制的约束。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：

Ø 保真性：应用程序在虚拟机上执行，除了时间因素外（比在物理硬件上慢一些），将表现为与在物理硬件上相同的执行行为。

Ø 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虚拟机管理器不干预 的情况下，直接在物理硬件上执行。

Ø 安全性：物理硬件应该由虚拟机管理器全权管理，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。

服务器虚拟化就是使软件和硬件相互分离，把软件从主要安装硬件中分离出来。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。

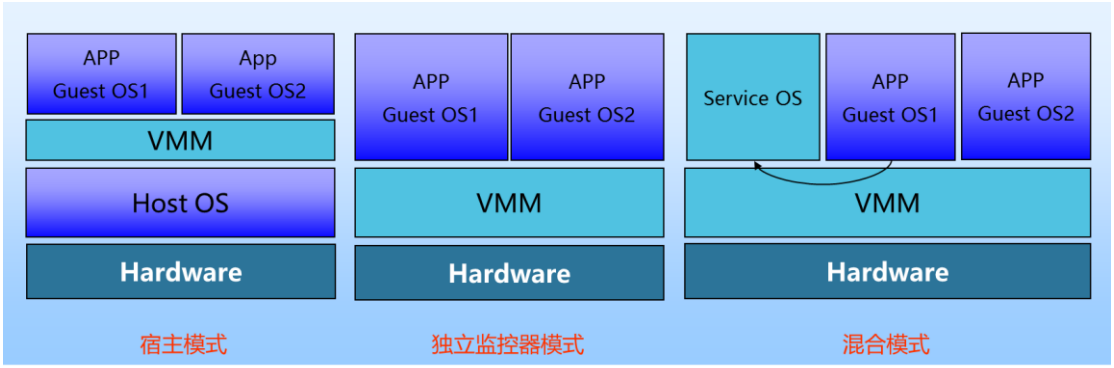


图 3.2-1 虚拟机监控器类型

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：

Ø **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,管理和实用底层硬件资源，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)。虚拟机操作系统(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。
- **分区隔离**：多个虚拟机可以同时运行，同时虚拟机之间还实现数据处理、网络连接和数 据存储的安全隔离。
- **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。

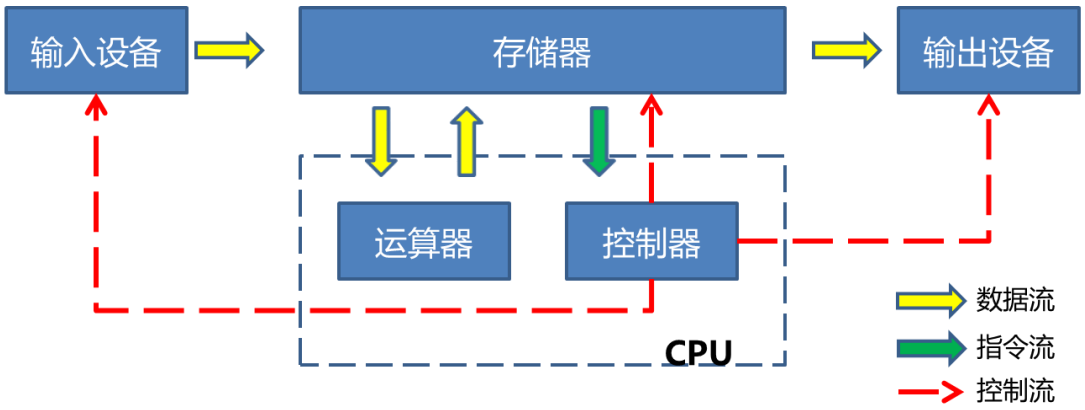


图 3.2-2 冯•诺依曼体系结构

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

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）决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。

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。

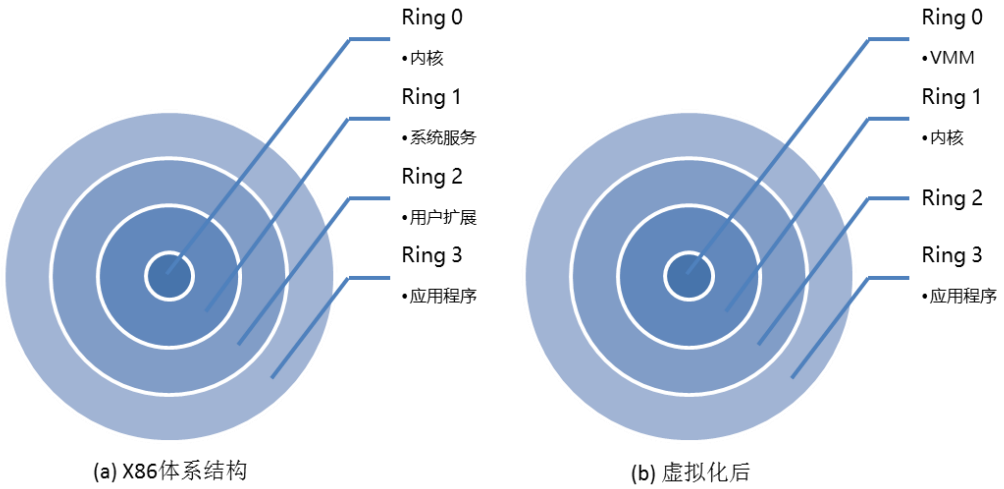


图 3.2.1-1 特权级

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”。

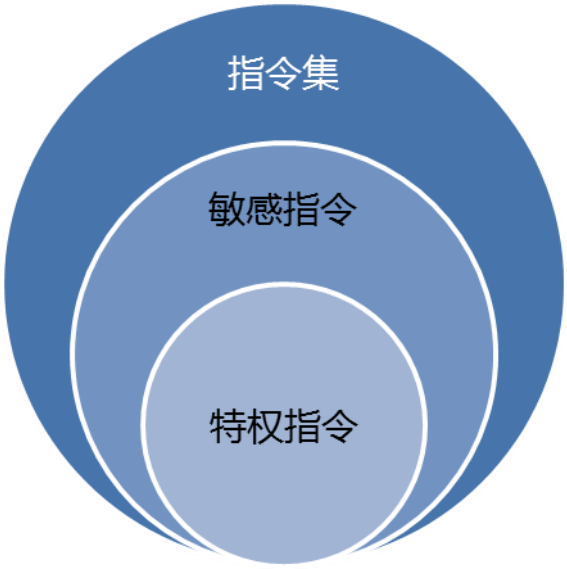


图 3.2.1-2 虚拟化漏洞

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

3.2.2 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：

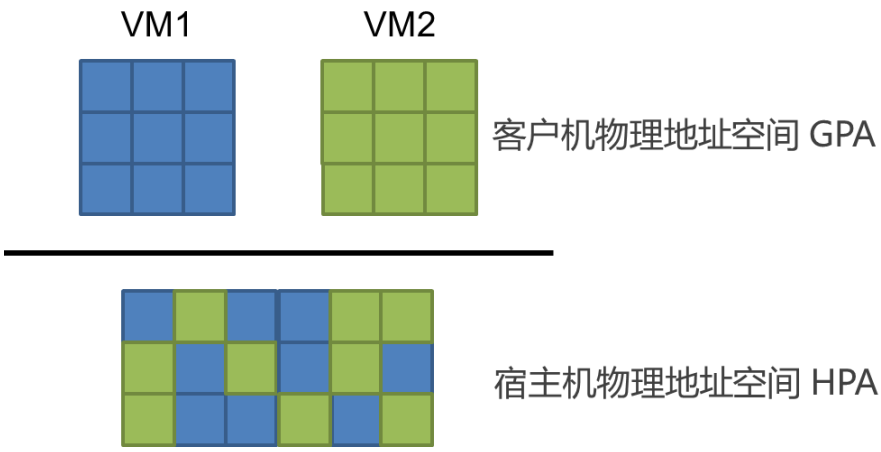


图 3.2.2-1 客户机物理空间地址

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，如图 3.2.2-2 所示：

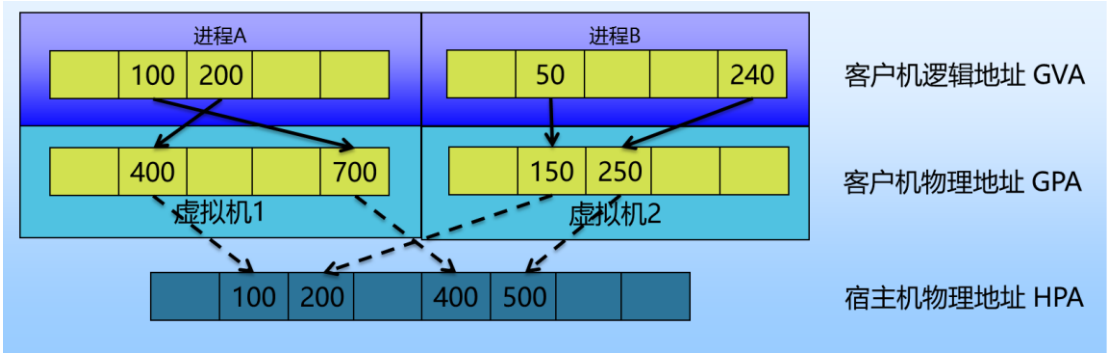


图 3.2.2-2 内存地址空间三层映射

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，Xen 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，如图 3.2.2-3 所示。

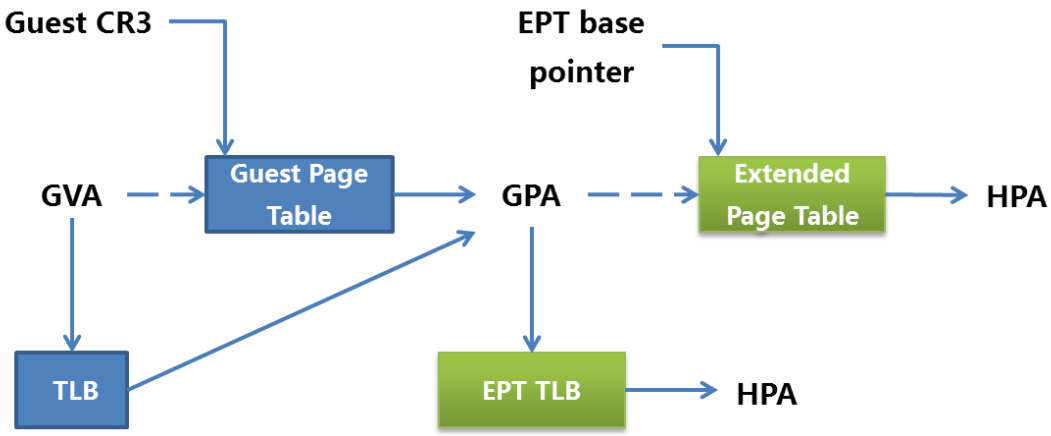


图 3.2.2-3 EPT 扩展页表

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，然后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：设备接口完全模拟、前端 / 后端模拟、直接划分。

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，Xen 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, PCI-SIG 之 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 第四章 InCloud Sphere 功能原理

通过阅读本章，您可以了解到：

- InCloud Sphere 4.5 旗舰版系统的主要功能；
- InCloud Sphere 4.5 旗舰版各功能的实现原理；

4.1 计算

4.1.1 CPU 管理

CPU 绑定

默认情况下 InCloud Sphere 旗舰版使用的是 Credit-Based CPU Scheduler，一种基于权重的 CPU 调度算法。简单的说就是给每个 VM 的 CPU 一个权重，权重多的得到和使用物理 CPU 的时间片较长，在 I/O 负载较重的情况下则会出现VM 的 I/O 等待，系统大量频繁的 swapping，从而影响服务器整体性能和其他用户。

因此为避免虚拟机“邻位干扰”，InCloud Sphere 旗舰版支持虚拟机预分配专享的物理 CPU 资源，同主机中其他虚拟机无法获取预分配的 CPU 资源，同时在运行过程中实际使用的 CPU 资源，允许随工作负载在分配的 CPU 资源范围内变动。

CPU 热添加

由于前期调研或规划不足导致 VM 的 CPU 资源紧缺，因此需要扩容 CPU 资源。若需扩容的 VM 运行是业务连续性要求比较高的系统，可以通过 CPU 热添加功能在不停机的情况下调整 CPU 资源。

4.1.2 内存管理

InCloud Sphere 旗舰版对 VM 内存管理分别采用固定和动态模式两种方式，固定内存是首次在创建 VM 时分配固定数量的内存资源，动态内存是根据预先分配的内存资源进行动态的调整和获取。要增大 InCloud Sphere 环境中物理内存的利用率，可以使用动态内存控制（DMC），这是一种能够在 VM 之间动态重新分配内存的内存管理功能。

图 4.1.2-1 是动态内存设置，VM 可根据动态内存分配范围内自动获取内存资源。

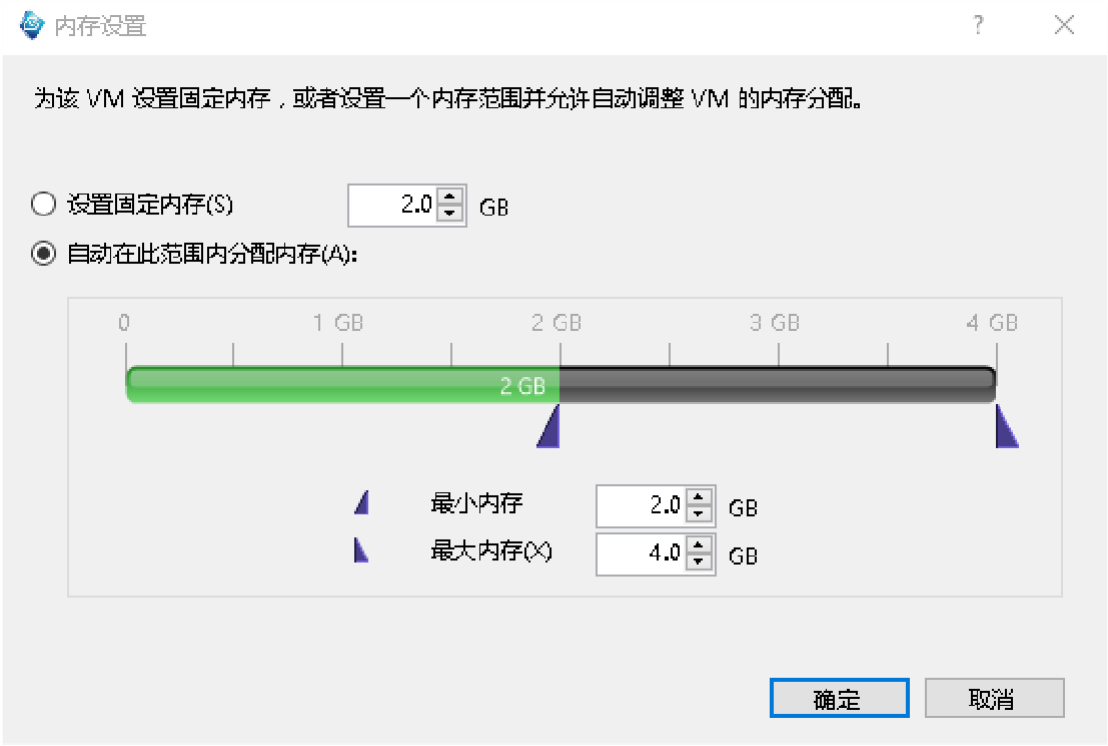


图 4.1.2-1 动态内存设置

InCloud Sphere DMC 工作原理

Ø 目标模式：管理员指定来宾操作系统的内存目标。InCloud Sphere 调整来宾操作系统的内存分配以满足目标的要求。在虚拟服务器环境中，InCloud Sphere 将调整来宾操作系统的内存分配，以满足指定的内存目标。

Ø 动态范围模式：管理员指定来宾操作系统的动态内存范围，InCloud Sphere 从该范围内选择一个目标，并调整来宾操作系统的内存分配以满足此目标。InCloud Sphere 动态重新分配主机内存以响应不断变化的来宾操作系统数目或不断变化的主机内存压力的任何情况下，指定动态内存范围特别有用。InCloud Sphere 从该范围内选择一个目标，并调整来宾操作系统的内存分配以满足该目标。

动态内存控制（DMC）优势

Ø 无需重新启动即可添加或删除内存，从而为用户提供更加优异的无缝体验；

Ø 服务器满载后，DMC 允许在服务器上启动更多的 VM，从而按照比例减少分配给正在运行的 VM。

4.1.3 GPU 管理

GPU 介绍

GPU 全称是 Graphic Processing Unit（图形处理器），其最大的作用进行各种绘制计算机图形所需的计算，包括顶点设置、光影、像素操作等。GPU 实际上是一组图形函数的集合，而这些函数由硬件实现。

InCloud Sphere GPU 管理

InCloud Sphere 包含两种 GPU 分配方式：breadth-first 和 depth-first，“breadth-first”这种分配机制是 GRID VGPU 在启动之后 VGPU 分配的最佳性能。针对 InCloud Sphere 如何选择两种模式？可通过 InCloud Sphere 主机属性中看到看中分配机制：

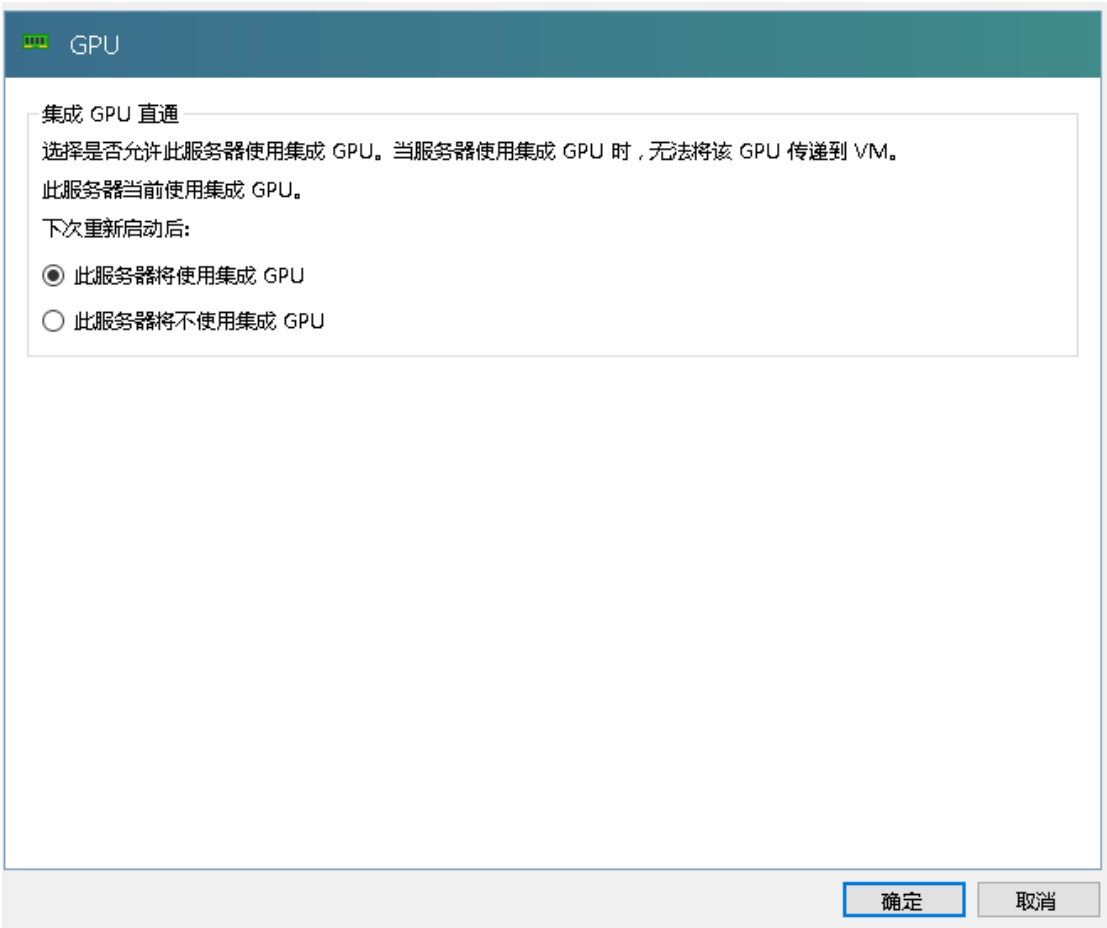


图 4.1.3-1 GPU 配置

Ø **此服务器将使用集成 GPU**：Depth-first 分配机制是默认设置，新启动的 VM 将尽可能的放置到一个已经使用了该vGPU profile 的 GPU 卡上，如果没有 GPU 使用该vGPU profile，则将尝试在一个空余的 GPU 启动虚拟机（如果有多余）。

Ø **此服务器将不使用集成 GPU**：新启动的虚拟机尽可能的防治到空余的 GPU 上，这样将获取最佳的性能，但是如果使用多个不同的 VGPU profiles，则可能会遇到问题。比如：如果有 2 台虚拟机使用 K220Q profile，则都会请求并分配到各自一个独立的 GPU 上，

如果后面有不是 K220Q profile 的虚拟机启动，则无法分配 GPU 导致无法启动，因为没有空余的 GPU 了，而 profile 又不能混合使用。

InCloud Sphere vGPU 工作原理

GPU 对于通用计算机和图形处理的内部组件主要有两部分：顶点处理器（vertex process）和子素处理器（fragment processor）。这种处理器具备流处理机的模式，即不具有大容量的快存/存储器可以读写，只是直接在芯片上利用临时寄存器进行流数据的操作

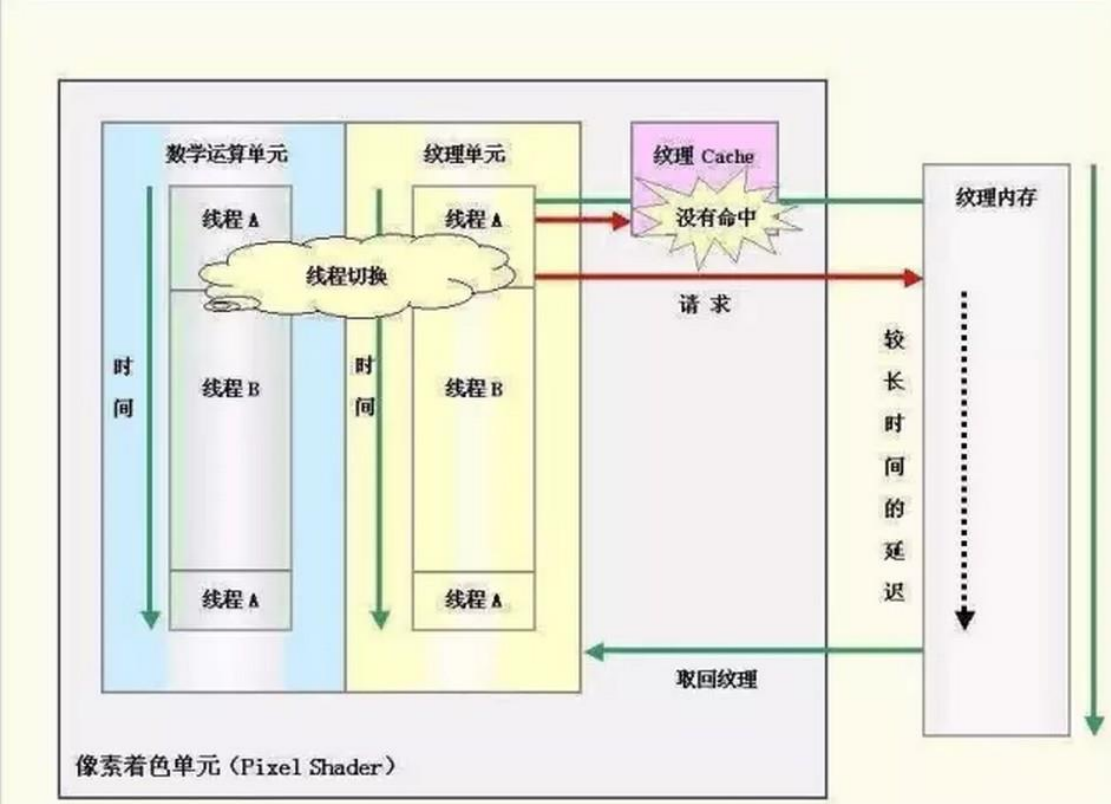


图 4.1.3-2 GPU 工作原理

当 GPU 用于图形处理是，此时 GPU 内部的顶点渲染、像素渲染以及集合渲染操作都可以通过流处理器完成。从图中可以看出，此时 GPU 内部的所有流处理器相当于一个多核的处理器，数据可以很方便的在不同的流处理器之间的输入和输出之间移动，同时 GPU 分派器和控制逻辑可以动态的指派流处理器进行相应的顶点，像素、几何等操作，因为流处理器都是通用的。

InCloud Sphere GPU 虚拟化

InCloud Sphere GPU 虚拟化就是将显卡进行切片，显卡时间分配给虚拟机使用的过程。由于支持显卡虚拟化的显卡一般可以根据需要切分成不同的规格时间片，因此可以分配给多台虚拟机使用。其实现原理其实是利用应用层接口虚拟化（API Remoting），API 重定向是指

在应用层进行拦截与GPU 相关的应用程序编程接口（Application Programming Interface、API），通过重定向（仍然使用 GPU）的方式完成相应功能，再将执行结果返回应用程序。

目前 GPU 虚拟化大部分是使用 NVDIA 公司提供的虚拟化技术，即是 vCUDA 技术。其 vCUDA（Virtual CUDA）包括三个模块：CUDA 客户端、CUDA 服务器端和 CUDA 管理端。客户端功能是在用户层提供针对 CUDA API 的库以及一个维护 CUDA 相关硬件状态的虚拟 GPU（vGPU）。服务器端组件位于特权虚拟机（InCloud Sphere 术语：特权域）中的应用层。特权虚拟机可以直接与硬件交互，因此服务器端组件可以直接操纵物理 GPU 来完成通用计算任务。管理端位于特权域（InCloud Sphere 术语），在实现 CUDA 编程接口虚拟化的基础上，将 GPU 强大的计算能力和计算资源在更高的逻辑层次上进行隔离、划分、调度。

InCloud Sphere 中运行 VMM 用于向上提供硬件映像，在 VMM 上运行若干个虚拟机，其中一个 VM 以特权虚拟机（Host VM）,即为 InCloud Sphere 中的 Domain 0,在虚拟机运行的操作系统称为 Host OS。Host OS 能够直接控制硬件，系统内安装着原生的 CUDA 库以及 GPU 驱动，所以Host OS 可以直接访问GPU 和使用GUDA。其他VM 属于非特权VM（Guest VM）,Guest VM 运行的 Guest OS 不能直接操纵 GPU。

4.2 存储

4.2.1 存储 I/O

InCloud Sphere 最新版本采用轮询技术处理虚拟存储设备 I/O，从而实现了性能和扩展性的提升，在测试中相比非轮询方式虚拟存储 I/O 性能提升 50%。

轮询技术介绍

针对每个虚拟机磁盘，dom0 会在用户态分别为磁盘创建一个 tapdisk3 进程，该进程主要负责虚拟机磁盘和 dom0 内核之间的数据和命令的交互。总的来说，它是一个单线程处理流程，以“监听-处理”的运行方式循环进行。

轮询是另一种处理方式，tapdisk3 循环地检查环以获取新的请求。在 InCloud Sphere 中 tapdisk3 负责处理虚拟存储设备 I/O，采用轮询的技术以提升其性能。在轮询方式中 tapdisk3 循环地检查以获取新的请求，不需要等待消息通道中断达到避免消息驱动带来的延迟，因此 tapdisk3 可以更迅速处理 I/O 请求。图 4.2.1.-1 是 I/O 请求图：

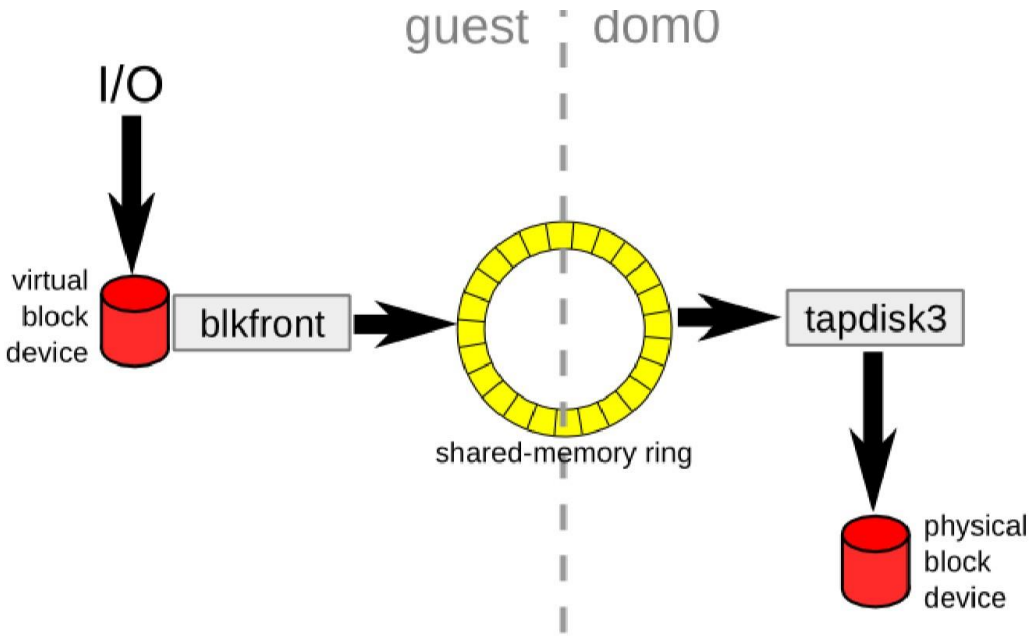


图 4.2.1-1 I/O 请求图

轮询技术用于降低时间驱动系统的延迟，要提升 I/O 性能，必须及时处理 I/O 请求，降低延迟是保证较低虚拟化开销的关键。因为物理 I/O 设备越来越快，任何在虚拟化层面的延迟都会越来越显著并导致吞吐率的降低。从虚拟机到物理存储设备的 I/O 的请求路径很长，tapdisk3 采用轮询可以优化此路径。

4.2.2 快照

InCloud Sphere 4.5 旗舰版支持磁盘快照、磁盘和内存快照两种快照方式：

Ø **磁盘快照**：存储 VM 的配置信息（元数据）和磁盘（存储），并允许导出和还原这些信息以作为备份。

Ø **磁盘和内存快照**：除了保存 VM 的元数据和磁盘外，还保存 VM 的内存状态。还原到磁盘和内存快照不需要重启 VM。生成快照时 VM 可处于运行或挂起状态。

虚拟机快照技术

两种经典的快照技术，COW（写时复制 Copy-On-Write）和 ROW（写重定向 Redirect-On-Write）：

Ø COW 最大的问题是对写性能有影响。第一次修改原卷，需要复制数据，因此需要多一次读写的数据块迁移过程。这个就比较要命，应用需要等待时间比较长。但原卷数据的 布局没有任何改变，因此对读性能没有任何影响。

Ø ROW 在传统存储情况下最大的问题是对读性能影响比较大。ROW 写的时候性能基本没有损耗，只是修改指针，实现效率很高。但多次读写后，原卷的数据就分散到各个地方， 对于连续读写的性能不如 COW。

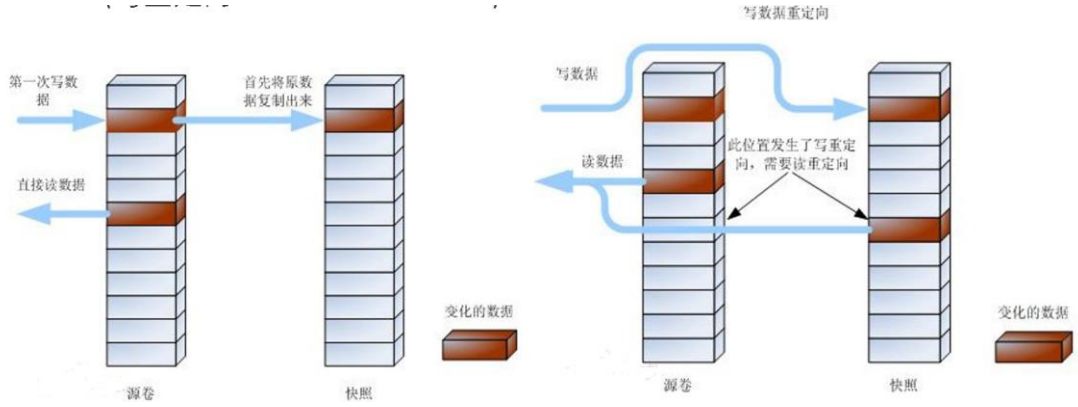


图 4.2.2-1 快照技术

4.2.3 存储多路径

功能描述

存储多路径是实现 InCloud Sphere 服务器和存储系统之间的路径选择和通信，同时还具有容错、负载及服务器群集功能。因两个路由通道上都具有活动通信，因此默认情况下 InCloud Sphere 使用循环模式负责平衡。InCloud Sphere 通过 iCenter 和 CLI 启用多路径。建议生产环境必须启用多路径功能。图 4.2.3-1 是 InCloud Sphere 多路径逻辑链路图：

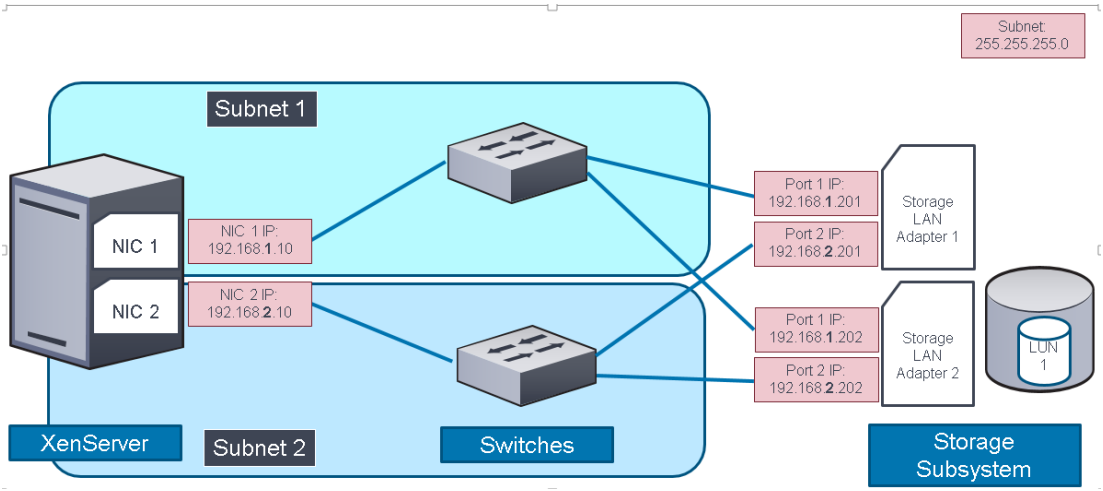


图 4.2.3-1 InCloud Sphere 多路径逻辑链路图

启用存储多路径优势：

Ø 路径冗余存储，InCloud Sphere 服务器可以选择最优的路径与存储系统通信

Ø 提高链路冗余，当一条链路故障时可自动切换到备用链路上

Ø I/O 负载均衡功能，作用是分担网络流量或扩展贷款，实现由多路径承载到达同一个目的地的 I/O 流量

4.2.4 存储读缓存技术

读取缓存可以改进 VM 的磁盘性能，因为从外部此案进行初始读取后，数据魂村在主机的可用内存中。在单一基础 VM 上克隆多个 VM 的情况下，它可以显著改进性能。例如在批量创建 VM 环境中，它将显著减少从磁盘读取的块的数量。

无论何时需要从磁盘多次读取数据，都可以看到这种性能改进，因为数据缓存在内存中。 例如，当引导风暴时或者大量 VM 同时运行恶意软件扫描（杀毒风暴）时。

对于基于文件的 SR，例如 NFS 和 EXT3 SR 类型，读取缓存在默认情况下处于启用状态。它针对所有其他 SR 处于禁用状态。

读取缓存适用于只读 VDI 和父 VDI，存在于从“快速克隆”或快照磁盘创建 VM 的情况中。从单一映像克隆 VM 时，将看到显著的性能改进。

性能改进取决于主机控制域（dom0）中的可用内存量，增加 dom0 的内存量将允许为读取缓存分配更多内存。

InCloud Sphere 允许设置缓存大小，通过 InCloud Sphere 的控制域（dom0）分配更多的内存，可以优化读取缓存性能。为实现优化，应当逐一在池中的全部主机上设置读取缓存大 小。此外，还必须在池中的所有主机上设置读取缓存大小的任何后续更改。

对读取缓存大小执行任何更改后，必须重新启动所有主机。

4.3 网络

4.3.1 网络虚拟化架构

InCloud Sphere 在网络方面没有开饭专门的网络对战模式，而是使用现有的网络堆栈模式来进行集成。目前 InCloud Sphere 的网络堆栈主要有两种：Bridge 和 Open vSwitch 模式。

Bridge 模式

Bridge 模式是通过一个虚拟的网桥设备来实现桥接，设备可以绑定若干个以太网口，从而进行桥接，此模式中 Bridge 即是 switch 同时具备 switch 的最基本的数据转发和记录 MAC

地址与端口的对应关系功能。InCloud Sphere 网络堆栈架构如图 4.3.1-1：

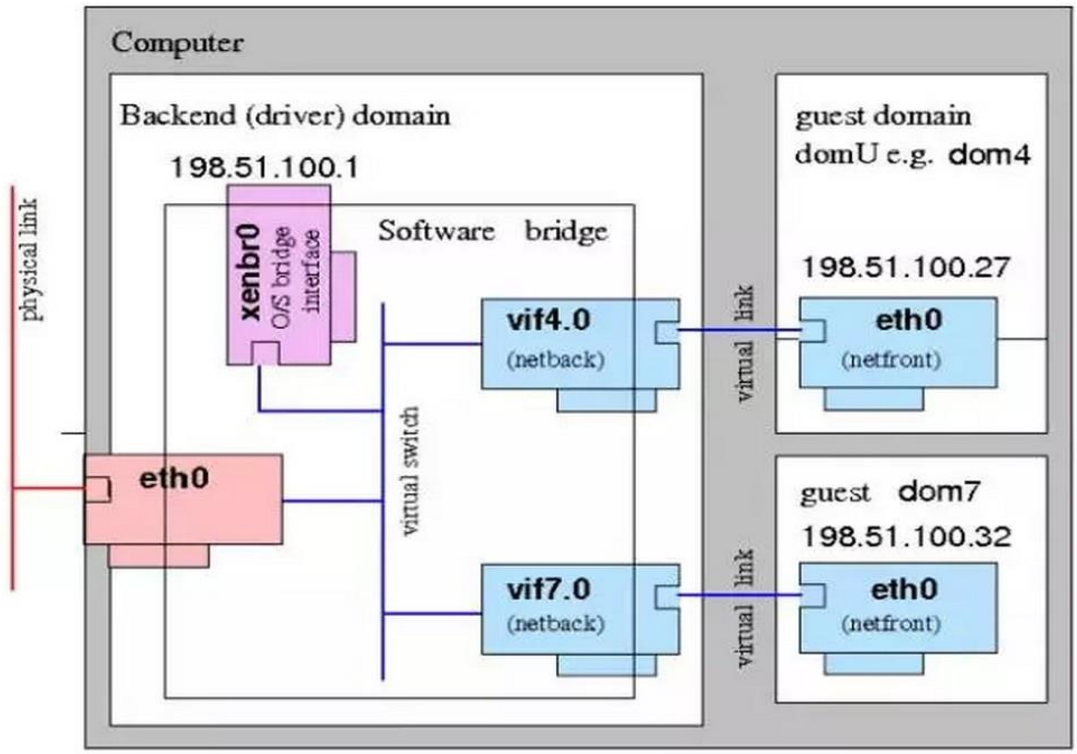


图 4.3.1-1 InCloud Sphere 网络堆栈架构

在 InCloud Sphere 中，一台物理服务器虚拟成多个 VM，VM 的 CPU、内存和 IO 设备都是虚拟设备，每台 VM 都有虚拟网卡。当所有的虚拟机需要和外部进行网络通讯时，必须使用服务器物理网卡进行实际的网络通讯，所以 VM 共享这张物理网卡的功能。而 InCloud Sphere 需要处理不同虚拟机的虚拟网卡和物理网卡之间的数据转发。

如图 4.3.1-1 所示 InCloud Sphere 会在 dom0 中创建 Bridge 设备，并且第一个 bridge 设备命令为（Bridge xenbr0），对于真实的物理网卡，称之为 peth#（physicaleth#）;同时 InCloud Sphere 在 dom0 中，为每个虚拟机的虚拟网卡都创建对应虚拟网络接口，并命名为 vifn.m（virtual interface.m）,其中 n 代表虚拟机对应的编号，m 代表了接口对应的虚拟机的虚拟网卡的编号。例如 vif0.0 对应 domain0 的的一块虚拟网卡 eth0，vif1.0 对应 domainU$#1 的第一块虚拟网卡 eth0。

图 4.3.1-1 中物理网卡 peth 和虚拟网卡接口 vif 都连接 Bridge，物理网卡负责和外部网络进行内部的数据包传递和转发。图中虚拟机 vif 分别桥接到 xenbr0 上，且桥接是工作在数据链路层，具体工作流程：

Ø 当外部的网络数据包到达物理网卡后，由 dom0 中的网卡驱动进行处理，

Ø Dom0 处理完成后传给 peth 接口，peth 接口将数据传递给 Bridge，即 xenbr0 设备

- xenbr0 设备的处理代码根据数据包的目的 MAC 地址判断报文该被转发到哪个虚拟接口 vif
- InCloud Sphere 虚拟化层处理 vif 与虚拟机中的虚拟网卡之间的数据传递工作
- 当 eth0 接收到的报文被提交给网桥的处理代码，Xen 虚拟化层判断报文转发、丢弃、或提交到协议栈上层

Open vSwitch 模式

Open vSwitch 是一款开源的软件虚拟交换机，实质上 InCloud Sphere 集成了该模式交换机，并对其进行大力支持。目前 InCloud Sphere 将其作为默认的网络堆栈模式。Open vSwitch 提供了比 Bridge 更多的功能，比如 VLAN、端口镜像、QOS 等。

Open vSwitch 和 InCloud Sphere 虚拟化底层结合方式，架构图如图 4.3.1-2 所示

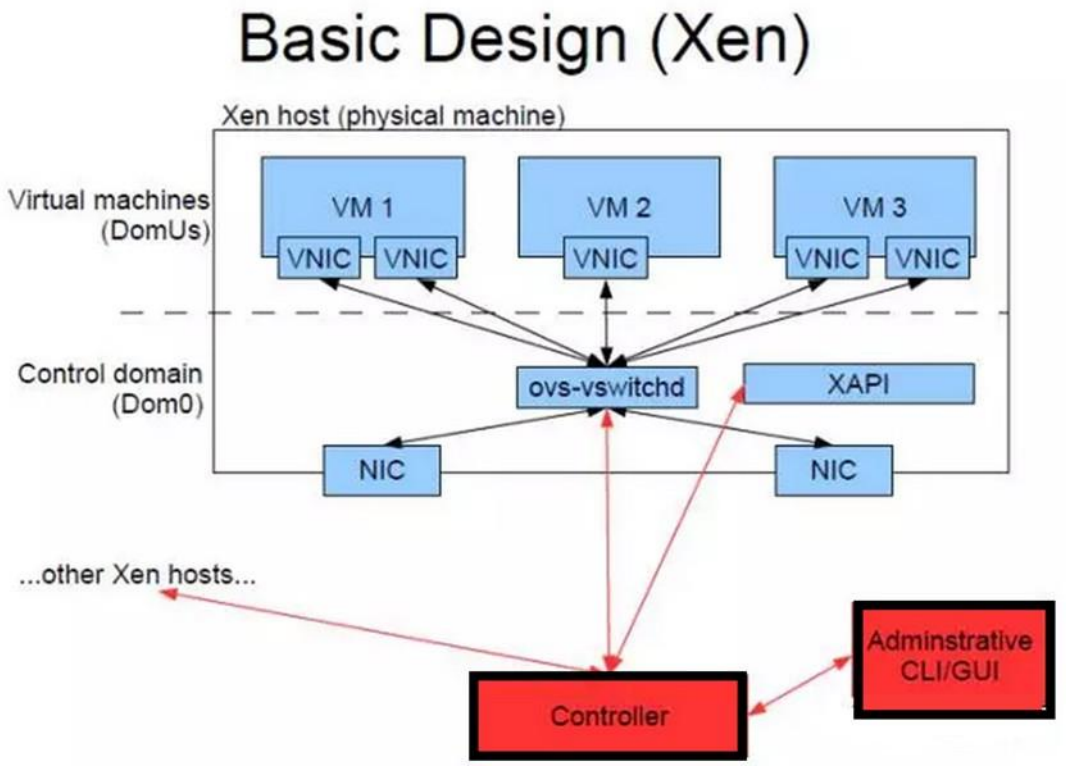


图 4.3.1-2 Open vSwitch 架构图

图 4.3.1-2 所示，Open vSwitch 在每台 InCloud Sphere 主机 Dom0 中创建数据路径 Datapath。然后通过统一的 controller 进行管理。

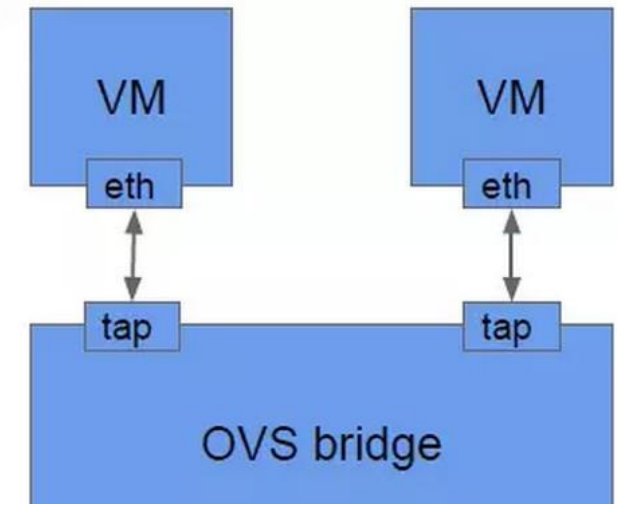


图 4.3.1-3 Open vSwitch 中的桥接

如图 4.3.1-3 所示，虚拟网卡与 Open vSwitch 的 Datapath 即网桥端口直接链接，数据包转发到 Datapath 之后由相应的处理机制进行数据包的处理。图中 tap 即是交换机上的一个端口。

4.3.2 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；且所有绑定的 NIC 共享同一个 MAC 地址。

InCloud Sphere 支持最多八个绑定网络，支持绑定模式有主动-主动、主动-被动和 LACP 模式。受支持网卡数量和受支持绑定模式因网络堆栈而异。

Ø LACP 绑定仅适用于 vSwitch，主动-主动和主动-被动适用于 vSwitch 和 Bridge 桥接

Ø 当 vSwitch 为网络堆栈时，可以将两个、三个或四个网卡绑定在一起

Ø 当 InCloud Sphere 的网络堆栈设置为 Bridge 模式时，只能绑定两个网卡

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路由 VM 通信，启用 Bridge 或 vSwitch 网络堆栈时，主动-主动是默认绑定模式。

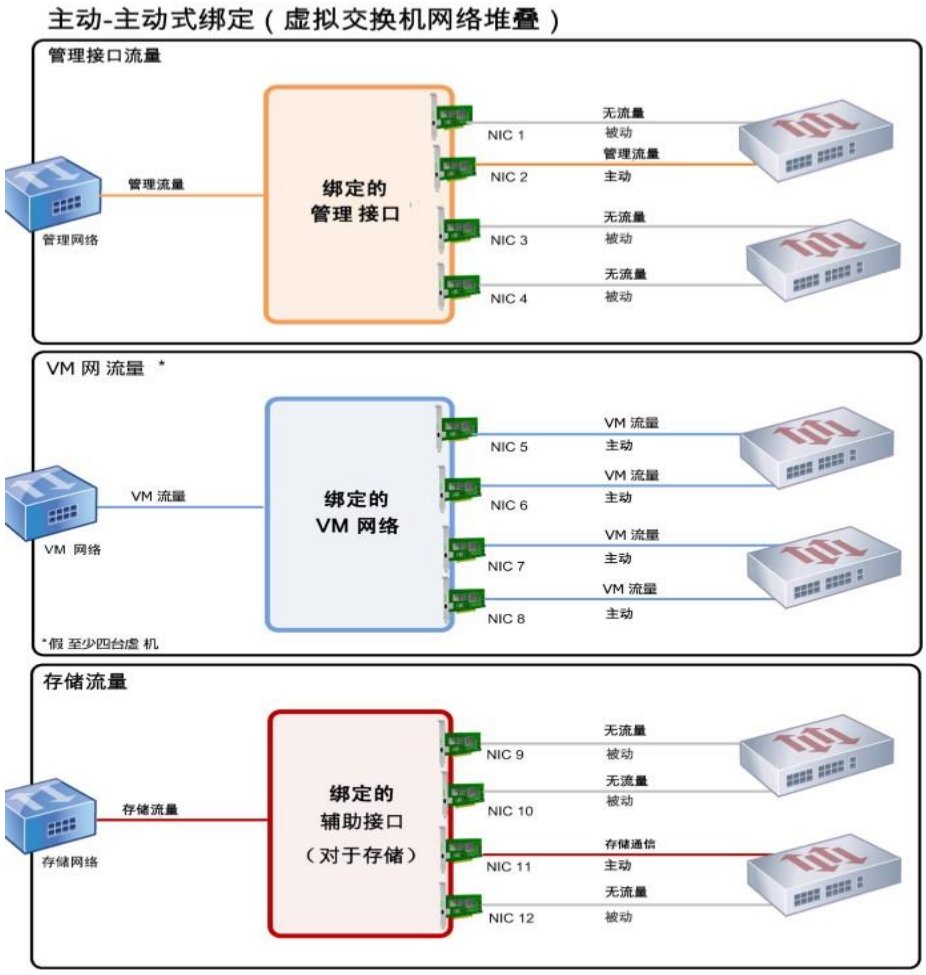


图 4.3.2-1 主动-主动绑定

如图 4.3.2-1 在管理网络中，NIC2 处于主动状态，NIC1、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 NIC11 处于活动状态，其他 NIC 无流量。

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。

图 4.3.2-2 显示了主动-被动模式下配置的两个绑定的 NIC

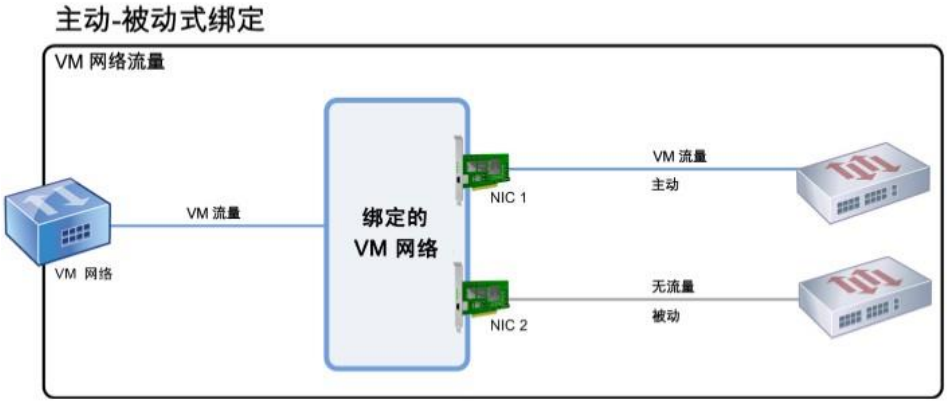


图 4.3.2-2 主动-被动绑定

图 4.3.2-2 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 NIC 之间移动。同样，主动-被动绑定可以配置两台交换机以实现冗余，但不需要堆栈。（如果管理交换机停机，堆栈交换机会成为单点故障）。

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：

Ÿ 基于源和目的地址负载均衡

Ÿ 基于 MAC 地址的负载均衡

基于源和目的地址的负载均衡：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 4.3.2-3，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

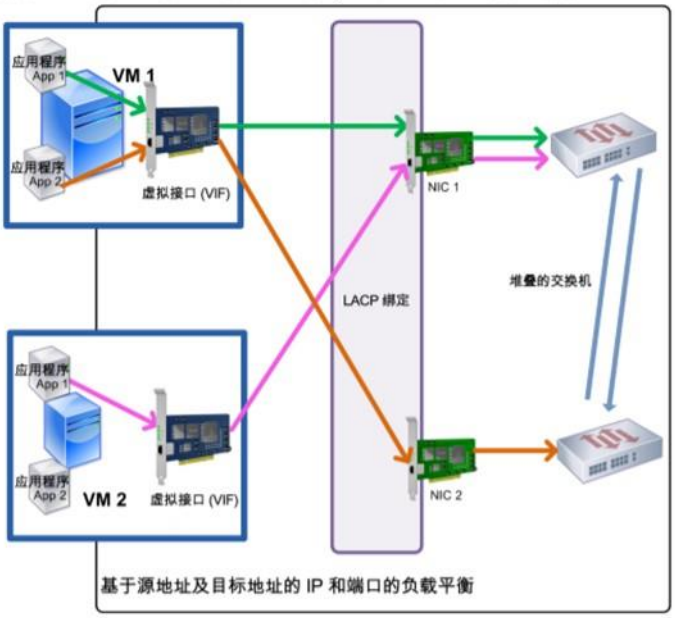


图 4.3.2-3 基于源和目的地址的负载均衡

基于源 MAC 地址的负载平衡：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。

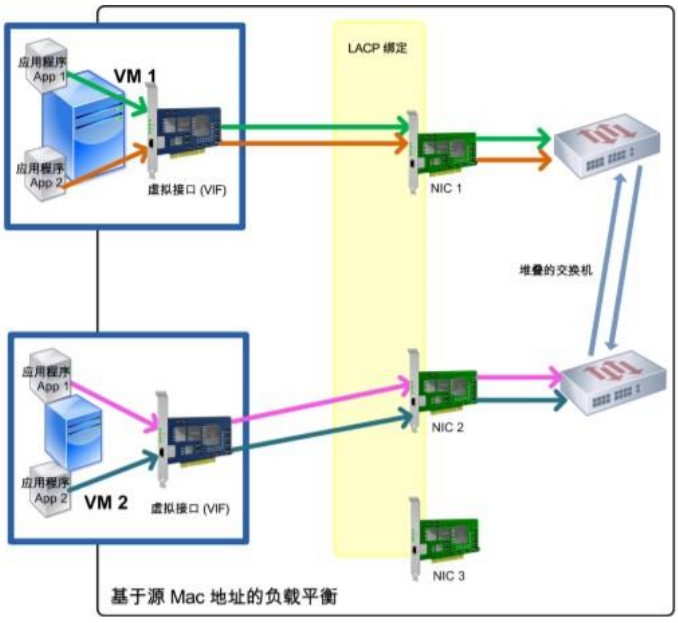


图 4.3.2-4 基于源 MAC 地址的负载平衡

图 4.3.2-4 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 VIF 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。

4.3.3 QoS

要限制 VM 每秒可以发送的传出数据量，可以针对 VM 虚拟接口 (VIF) 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（以每秒千字节为单位）。QoS 值将限制源自 VM 的传输速率。QoS 设置不会限制 VM 可以接收的数据量。如果需要限制接收量，浪潮建议您限制网络中较高层（例如，交换机层）传入数据包的速率。 根据在池中配置的网络堆栈，可以按照下表中的说明，在以下两个位置之一针对 VM 虚拟接口 (VIF) 设置服务质量 (QoS) 值：a) 在 vSwitch 控制器上；b) 在 InCloud Sphere 中（使用 CLI 或 iCenter）。

4.4 高可用

4.4.1 vMotion

vMotion 是 InCloud Sphere 旗舰版重要功能，vMotion 提供 VM 灵活性和可用性，可将正在运行的整个虚拟机能够在瞬间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。如图 4.4.1-1

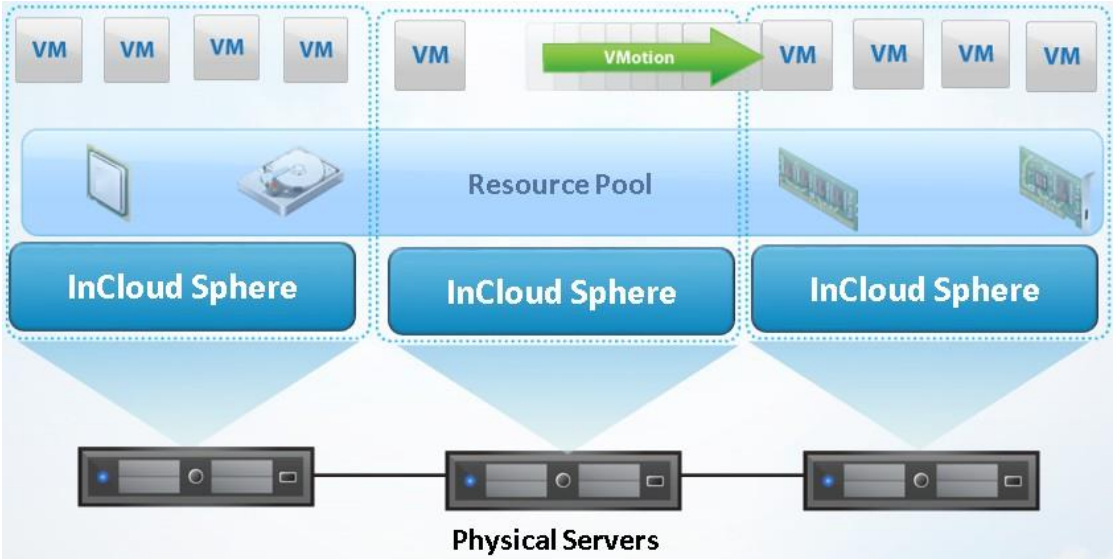


图 4.4.1-1 vMotion 示意图

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 迁移原理

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，图 4.4.1-2

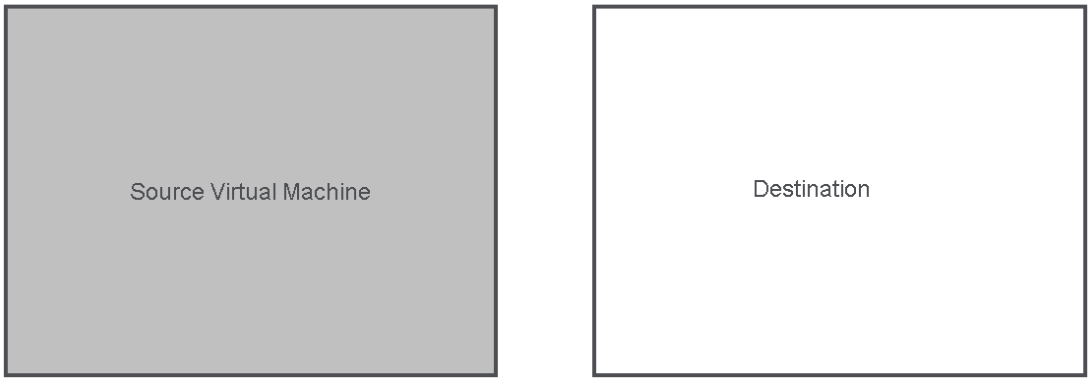


图 4.4.1-2

（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，图 4.4.1-3

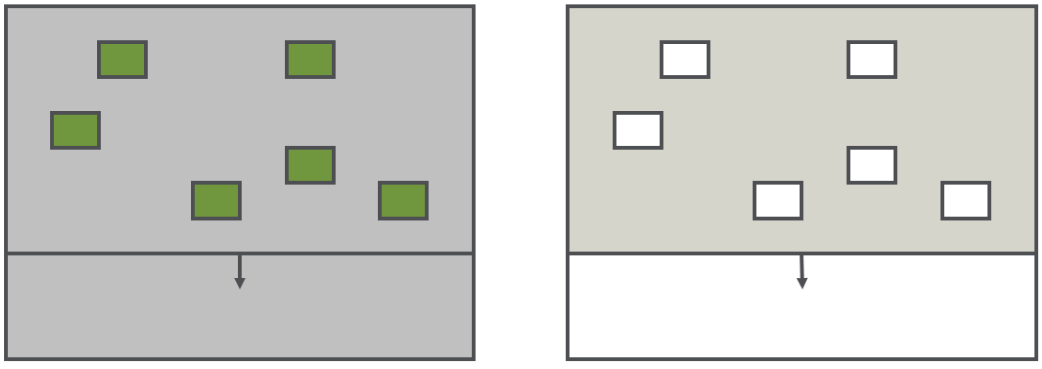


图 4.4.1-3

（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，图 4.4.1-4

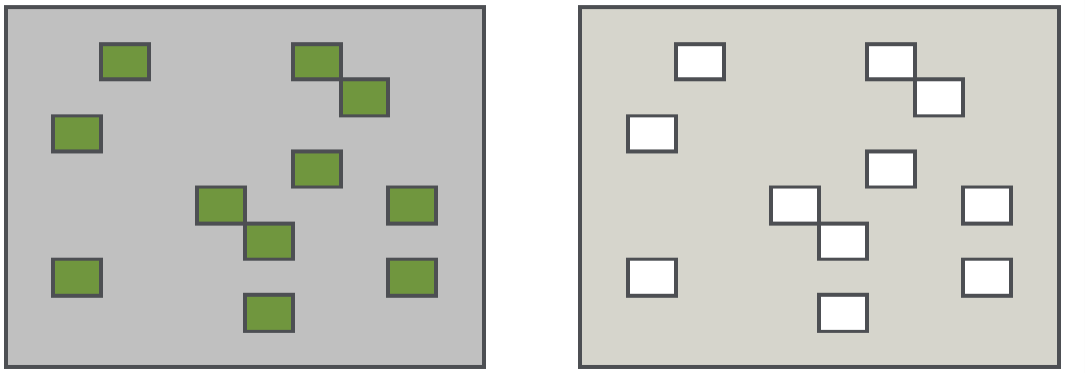


图 4.4.1-4

（4）加入发生了变化，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，图 4.4.1-5 和 4.4.1-6

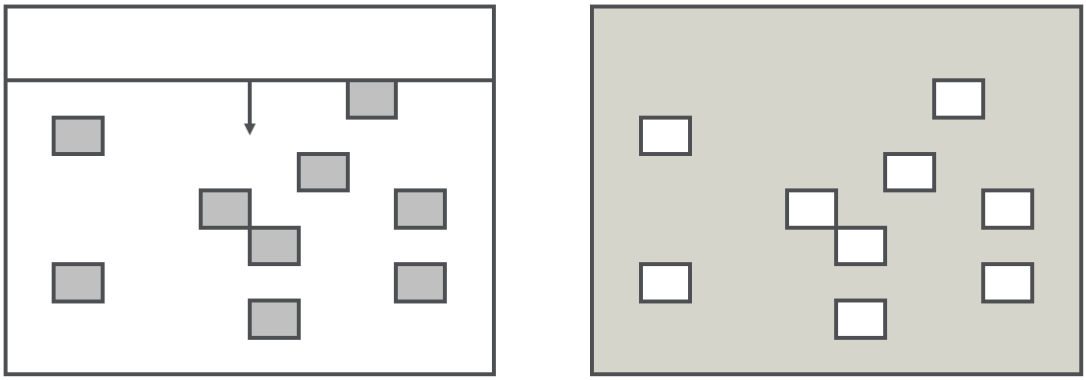


图 4.4.1-5

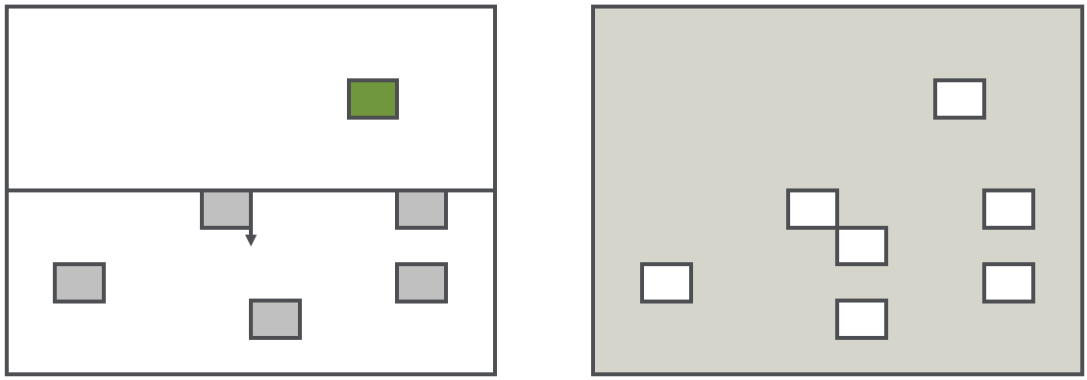


图 4.4.1-6

（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，而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，图 4.4.1-7

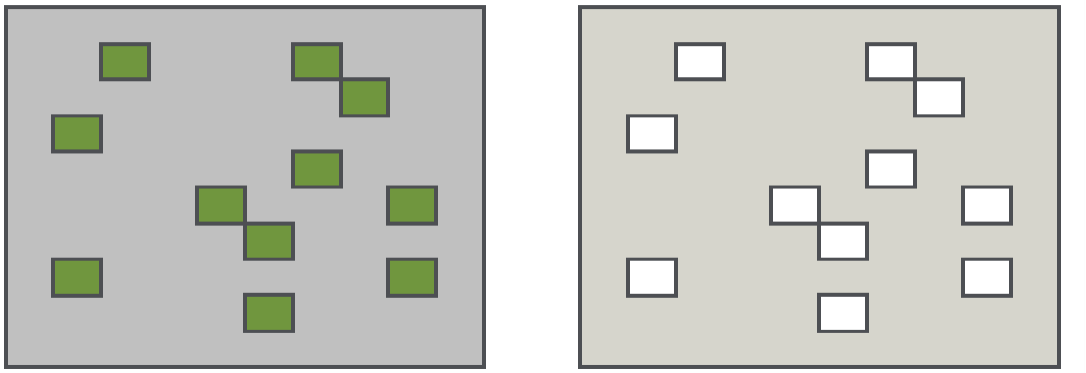


图 4.4.1-7

（6）内存复制完成之后，将 VM 的工作状态复制到目标服务器上，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。图 4.4.1-8

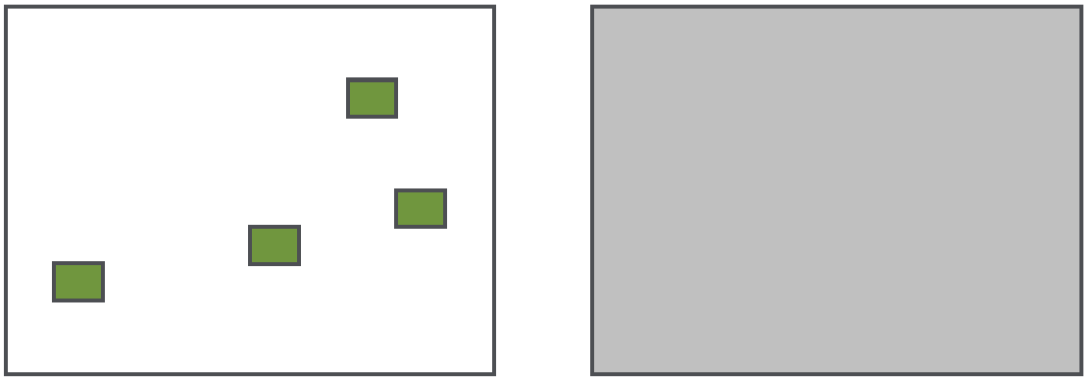


图 4.4.1-8

4.4.2 Storage vMotion

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 工作原理

（1）“A”为当前虚拟机的 VDI，在目的存储建立相同的“A”VDI 结构，并不包括数据，图 4.4.2-1

图 4.4.2-1

（2）在源上对“A”做一个快照“C”，快照“C”的结构也被复制到目的存储上，图 4.4.2-2

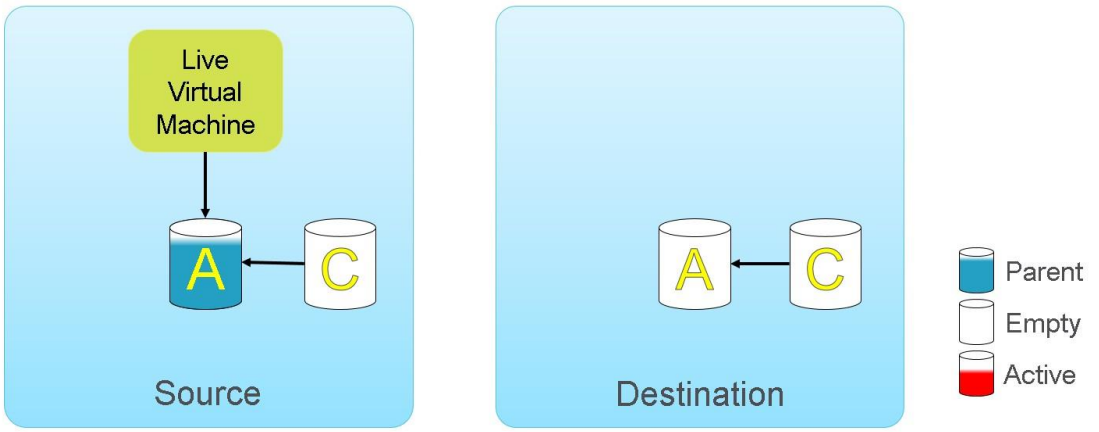


图 4.4.2-2

（3）当前虚拟机所有的数据同时写入源和目的的“C”，“A”的数据被复制到目标存储中，图 4.4.2-3

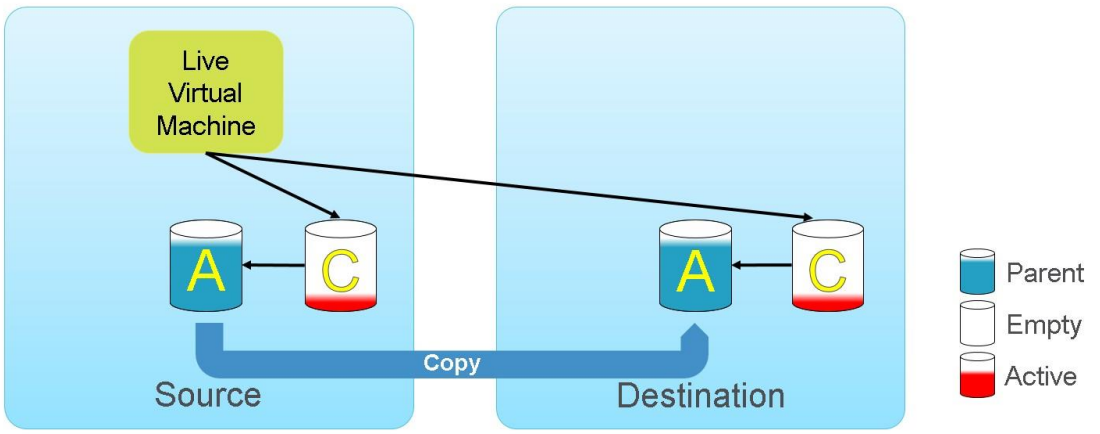


图 4.4.2-3

（4）当“A”复制完成后，开始进行虚拟机 vMotion 操作，“C”持续的同步写入持续进行 vMotion

操作结束，图 4.4.2-4

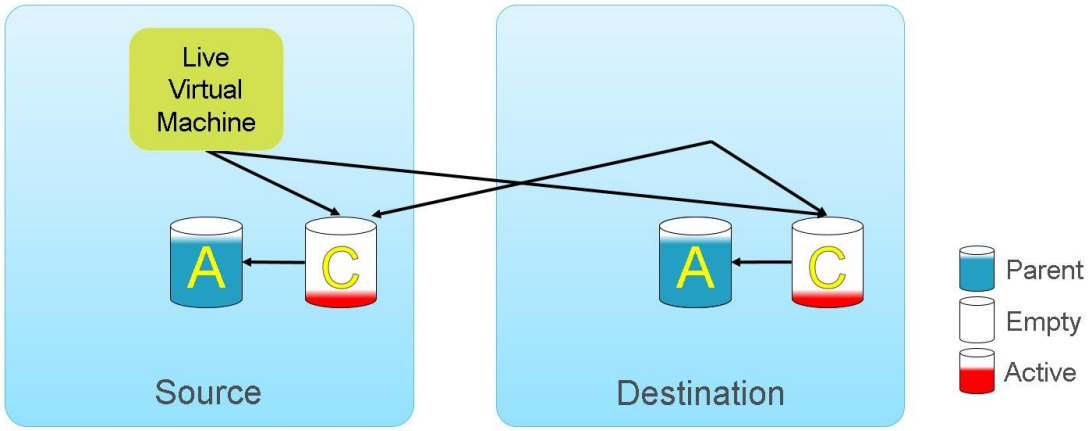


图 4.4.2-4

（5）资源进行释放，虚拟机的 Storage vMotion 完成，图 4.4.2-5

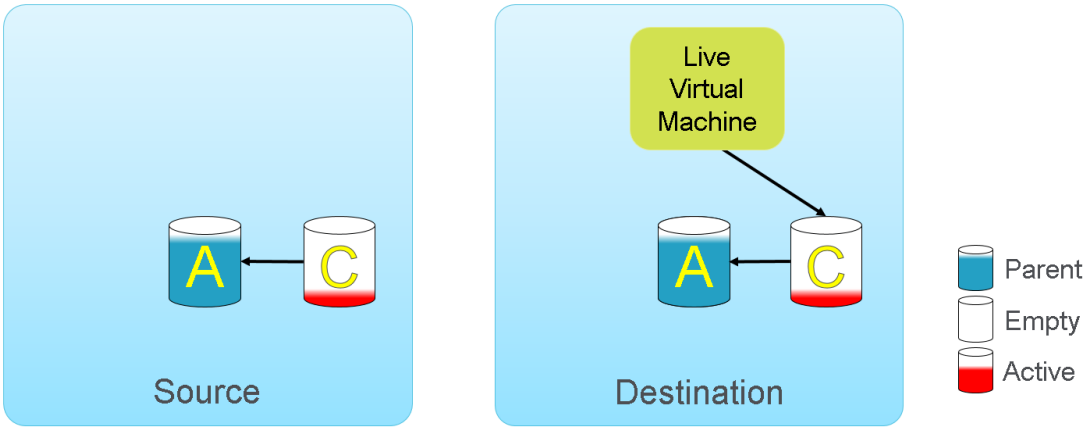


图 4.4.2-5

4.4.3 HA

InCloud Sphere 旗舰版 HA 是资源池中高级功能，为虚拟化的 IT 环境提供故障切换保护功能的组件。其功能允许资源池中出现故障 InCloud Sphere 主机承载的 VM 自动在其他主机上重新启动，其业务可在短时间内恢复正常。

如出现服务器故障（如网络连接断开）或发生控制堆栈问题，InCloud Sphere 主机将自我保护以确保 VM 没有同时在两台服务器上运行（若一个 VM 实例同时在不同的服务器上运行会导致 VM 数据丢失或损坏）。采取保护措施后，服务器立即重新启动，其所承载的 VM 停止运行；资源池中其他服务器检测到 VM 停止运行后，根据 HA 的配置策略重新启动 VM。

受保护的服务器将进入重新引导序列，并在重新启动后尝试重新加入资源池。

HA 工作原理

Ø 通过 iCenter 配置启用 HA，主节点调用 API 即 Poll.enable.ha，图 4.4.3-1

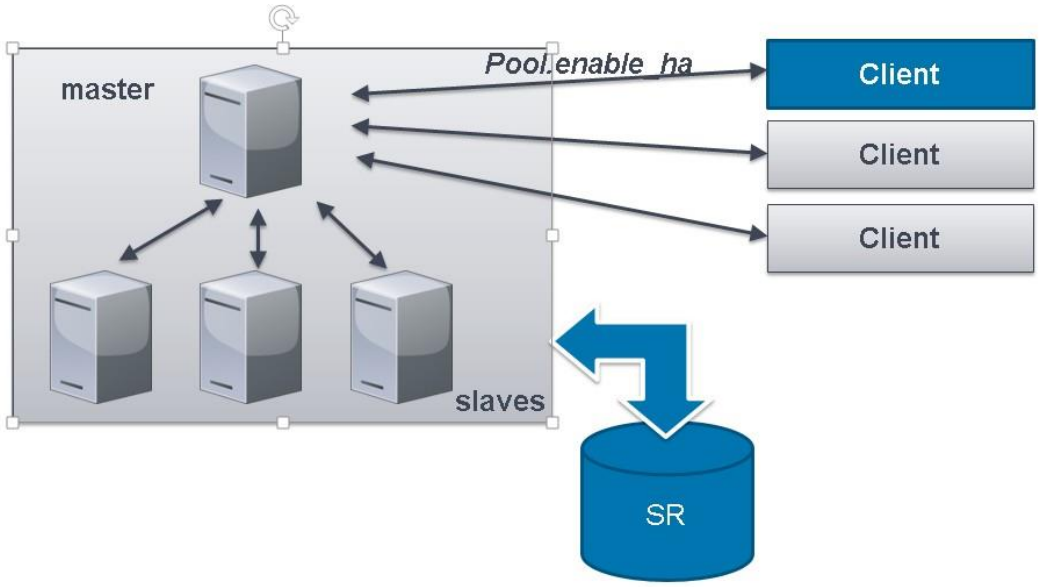


图 4.4.3-1

Ø 共享存储新建或复用原来的 HA 虚拟磁盘映像，保存主节点的元数据和状态信息，图 4.4.3-2

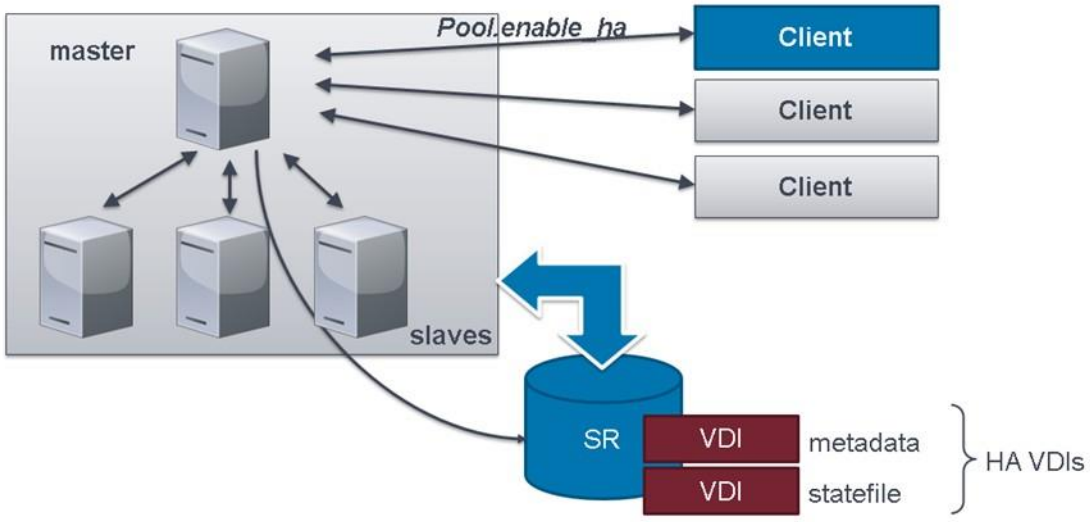


图 4.4.3-2

Ø 主节点保存元数据文件信息，池中所有主机接收状态文件信息并保存，防止主节点故障 时无可用的主机继续执行 HA，图 4.4.3-3

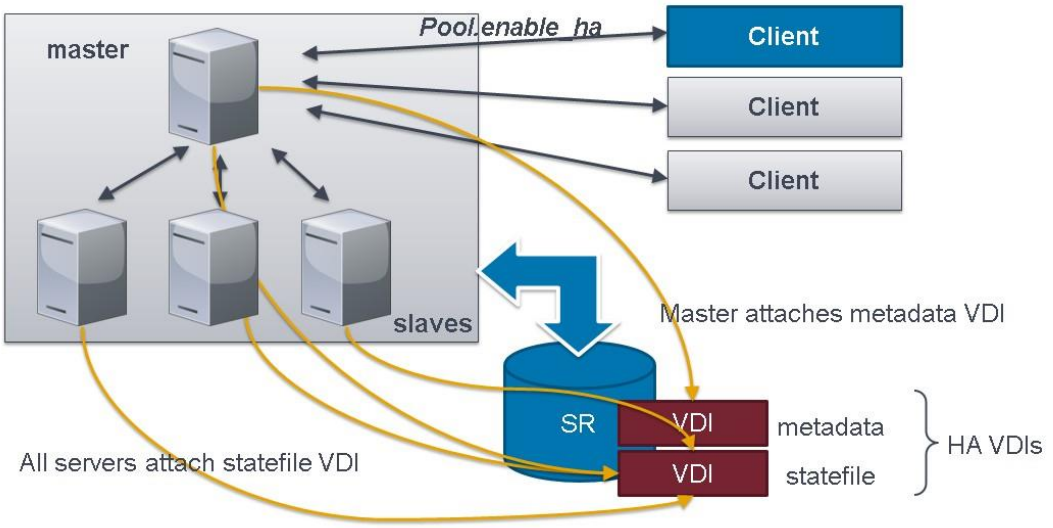


图 4.4.3-3

Ø 建立网络和存储的心跳监控状态，监控每台主机的运行状态，图 4.4.3-4

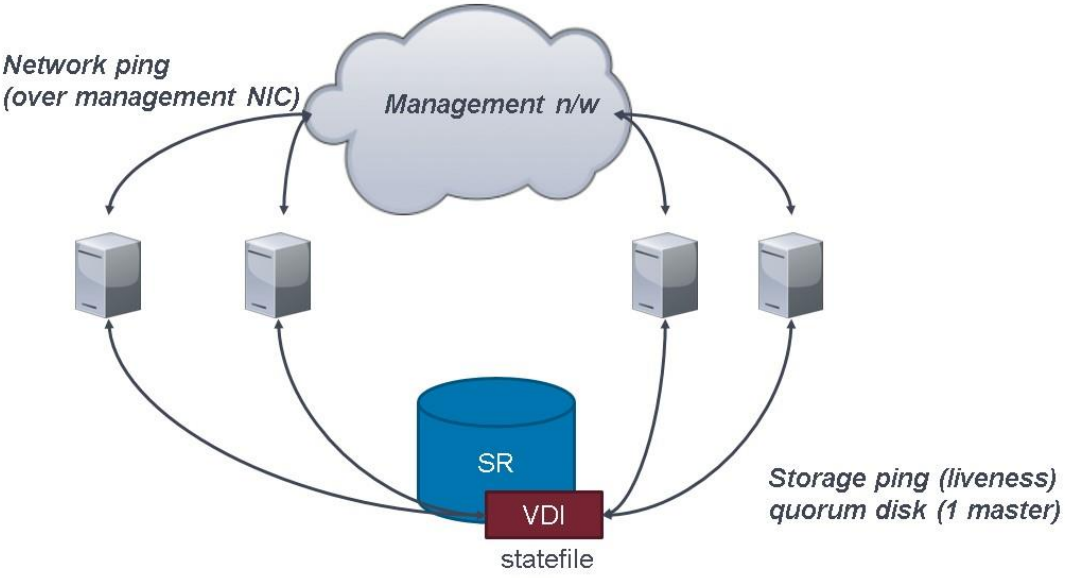


图 4.4.3-4

Ø HA 启动完成，主机网络和共享磁盘监控成功和错误状态信息，图 4.4.3-5

图 4.4.3-5

4.5 负载均衡

InCloud Sphere 负载均衡是以虚拟设备打包形式的组件（WLB，Workload Balancing），可用于：

Ø 创建关于 InCloud Sphere 环境 VM 性能的报告，图 4.5-1

Ø 评估资源利用情况，并根据工作负载将虚拟机放置在池中最合适的主机上

Ø 确定用来启动虚拟机的最佳主机

Ø 确定主机出现故障时虚拟机将移动到的最佳主机

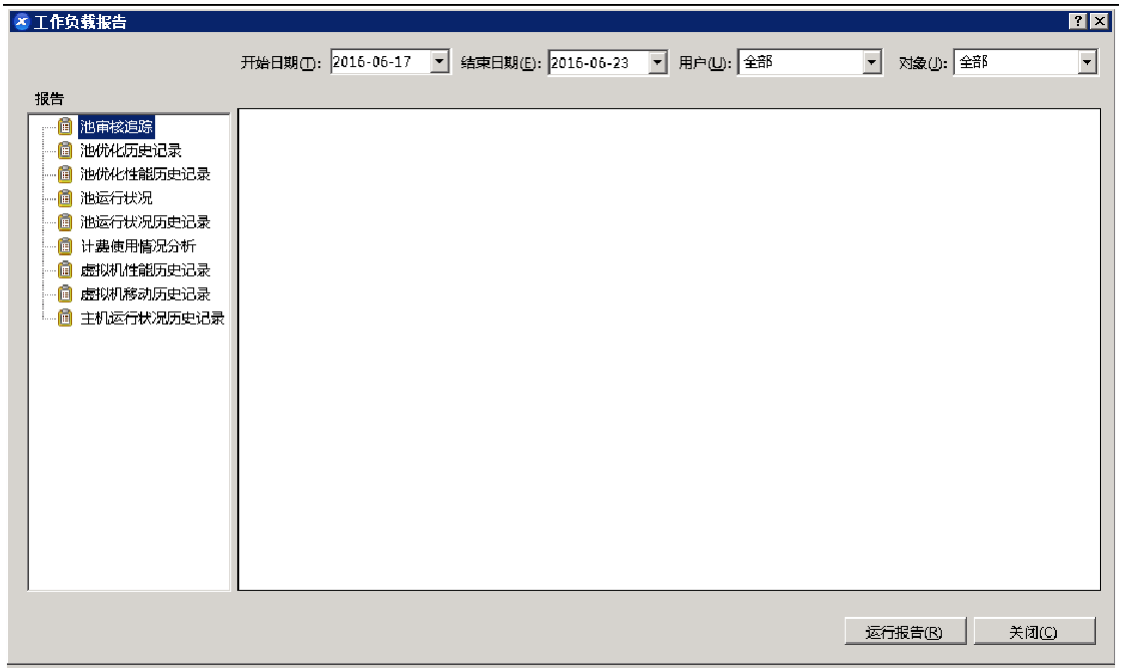


图 4.5-1 工作负载报告

可以导出的性能报告

优化模式

Ø 性能最大化（默认设置） Workload Balancing 尝试在资源池中的所有物理主机上平均分布工作负载。 目标是最大限度地降低所有主机上 CPU、内存和网络的压力。 Workload Balancing 将在某个主机达到高阈值时提出优化建议。

Ø 密度最大化，Workload Balancing 将尝试通过合并活动的虚拟机来最大限度地减少需要的物理主机数量。

Ø 如果选择密度最大化放置策略，Workload Balancing 将在某台虚拟机达到低阈值时提出合并优化建议。

Ø 可按照固定或时间段选择优化模式，同时可禁用或启用某个时间段的优化模式。

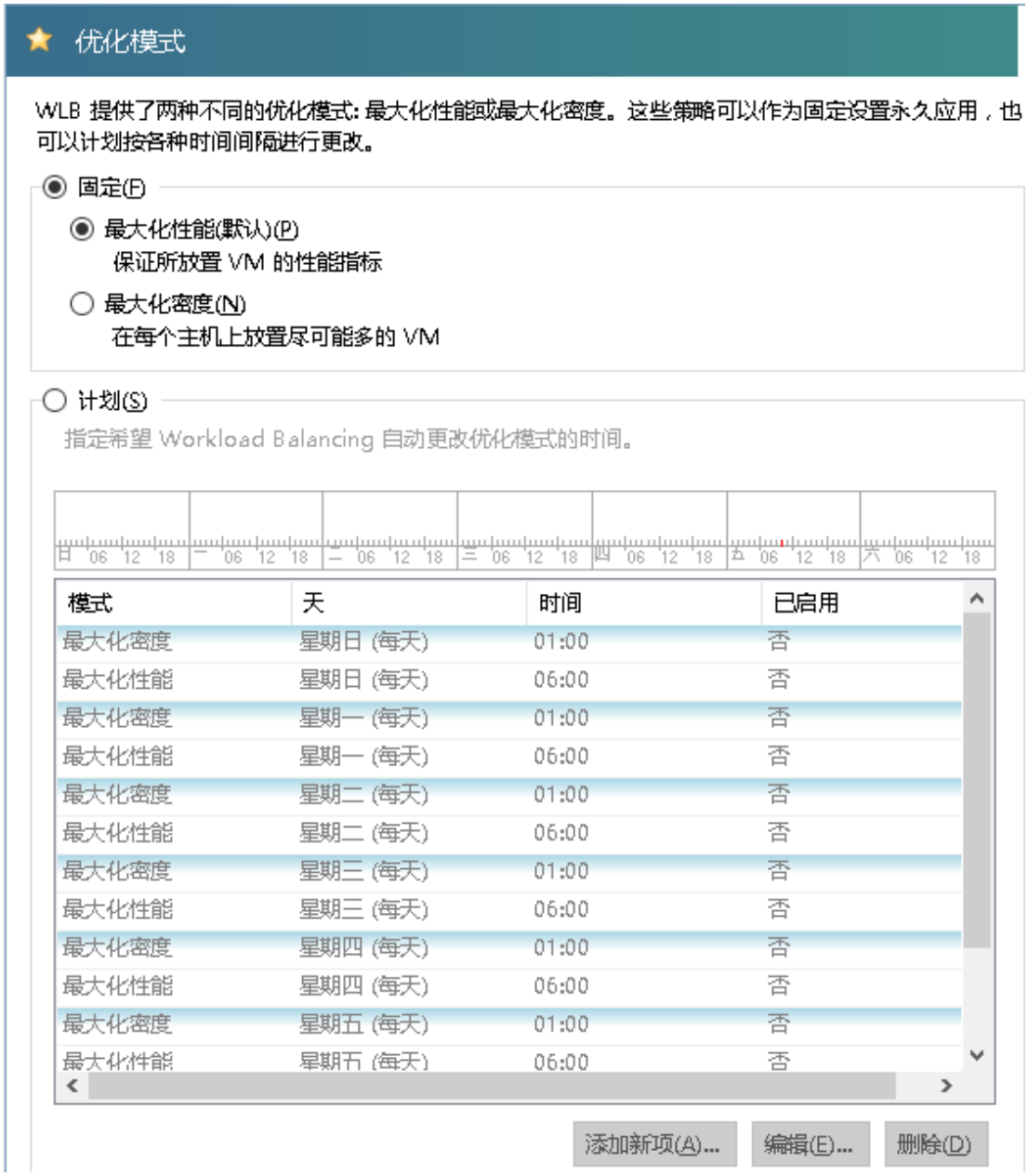


图 4.5-2

自动优化

Ø 可以将 Workload Balancing 配置为自动应用建议（自动化），并自动打开或关闭主机（电源管理）。

Ø 要自动关闭主机（例如在使用率较低的时段），必须 Workload Balancing 配置为自动应用建议并启用电源管理功能。

Ø 要在虚拟机使用率下降时关闭主机，必须配置自动化功能、电源管理功能和密度最大化 模式。

图 4.5-3

优化过程

（1）Workload Balancing 每两分钟对池中的各主机进行一次资源利用率评估

（2）如果主机上的资源利用率超出相关的阈值，Workload Balancing 将资源的当前利用率与三十分钟前和二十四小时前的历史数据结合起来，算出历史平均利用率。

（3） Workload Balancing 使用指标加权来确定应该首先优化的主机

（4） Workload Balancing 确定哪些主机可以支持迁出的 VM

（5）对主机和 VM 进行评估后，Workload Balancing 将尝试构建虚拟模型

4.6 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、C-state/P-state 信息以及存储。适用情况下，会针对每个主机和每个 VM 提供这些指标。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 xe CLI 或 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。

4.6.1 性能收集

客户可以使用通过轮询数据库 (RRD) 公布的指标，监视其 InCloud Sphere 主机和虚拟机(VM)的性能。这些指标可以在 HTTP 上或通过 RRD2CSV 工具查询。

部分可用主机指标（表 4.6.1-1）

表 4.6.1-1 部分可用主机指标

指标名称	说明	条件	ICS 名称
avgqu_sz_<sr-uuidshort>	平均 I/O 队列大小（请求）。	主机上的 SR<sr>中至少有一个插入的 VBD	<sr> 队列大小
cpu<cpu>-C<cstate>	CPU <cpu> 处于 C-state <cstate> 下的时间（以毫秒为单位）。	C-state 存在于 CPU 上	CPU <cpu> C-state <cstate>
cpu<cpu>-P<pstate>	CPU <cpu> 处于 P-state <pstate> 下的时间（以毫秒为单位）。	P-state 存在于 CPU 上	CPU <cpu> P-state <pstate>
cpu<cpu>	物理 CPU <cpu> 的利用率（片段）。默认情况下，此选项处于启用状态。	pCPU<cpu>存在	CPU <cpu>
cpu_avg	物理 CPU 平均利用率（片段）。默认情况下，此选项处于启用状态。	无	平均CPU 使用率

部分可用 VM 指标（表 4.6.1-2）

指标名称	说明	条件	ICS 名称
memory_internal_free	按来宾系统代理报告使用的内存(KiB)。默认情况下处于启用状态	无	可用内存
runstate_fullrun	所有 vCPU 运行的时间片段。	无	vCPU 完全运行
runstate_full_contention	所有 vCPU 的可运行时间片段（即，等待 CPU）	无	vCPU 完全争用
runstate_concurrency_hazard	有些 vCPU 正在运行和有些 vCPU 可运行的时间片段	无	vCPU 并发危险
runstate_blocked	所有 vCPU 受阻或脱机的时间片段	无	vCPU 空闲
runstate_partial_run	有些 vCPU 正在运行和有些 vCPU 受阻的时间片段	无	vCPU 部分运行

runstate_partial_contention	有些 vCPU 可运行和有些 vCPU 受阻的时间片段	无	vCPU 部分争用
vbd_<vbd>_write	每秒写入设备<vbd>的字节数。默认情况下处于启用状态	VBD <vbd> 存在	磁盘 <vbd> 写入
vbd_<vbd>_read	每秒从设备<vbd>读取的字节数。默认情况下，此选项处于启用状态。	VBD <vbd> 存在	磁盘 <vbd> 读取

4.6.2 配置性能图表

iCenter 界面性能监视图标分为“折线图”和“区域图”两种，同时可在监视界面中添加新的性能监视图标或编辑现有性能监视图标。

折线图

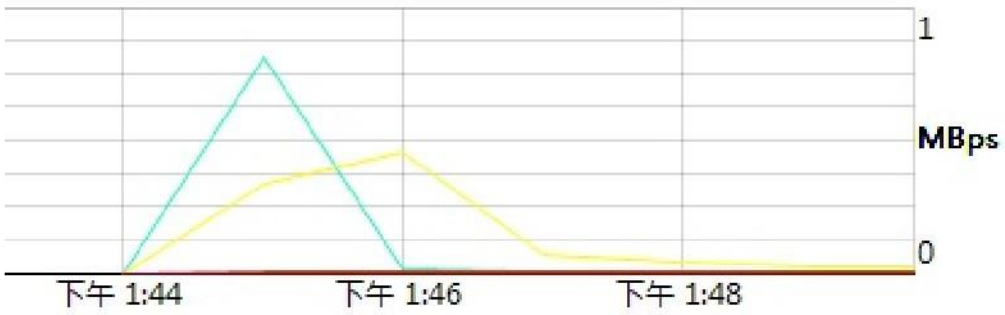


图 4.6.2-1

区域图

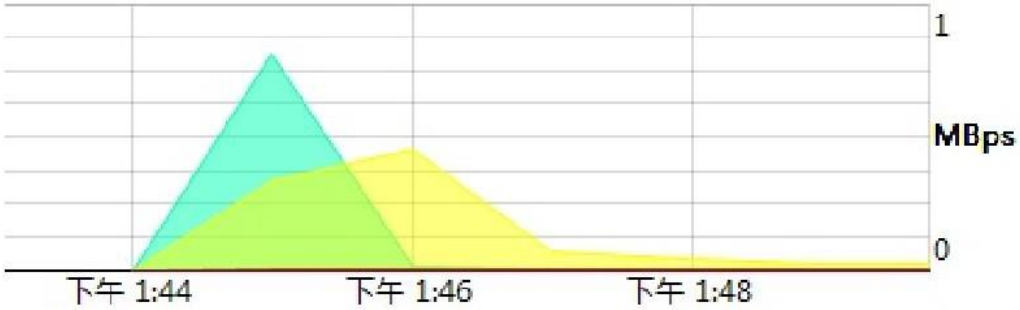


图 4.6.2-2

4.6.3 自动化告警机制

客户能够配置 InCloud Sphere，使其在 InCloud Sphere 主机生成警报时发送电子邮件通

知，可通过使用 iCenter 或使用 xe 命令行接口 (CLI) 完成。

InCloud Sphere 中的 mail-alarm 实用程序使用 SMTP 发送电子邮件通知。发送电子邮件通知之前，mailalarm 实用程序会查找配置文件 mail-alarm.conf，如果此配置文件存在，文件内容将用于配置 SMTP；否则，XAPI 数据库（使用 iCenter 或 xe CLI 配置）中的详细信息将用于发送电子邮件警报。要通过经过身份验证的SMP 服务器发送电子邮件通知，如图4.6.3-1 和 4.6.3-2

图 4.6.3-1

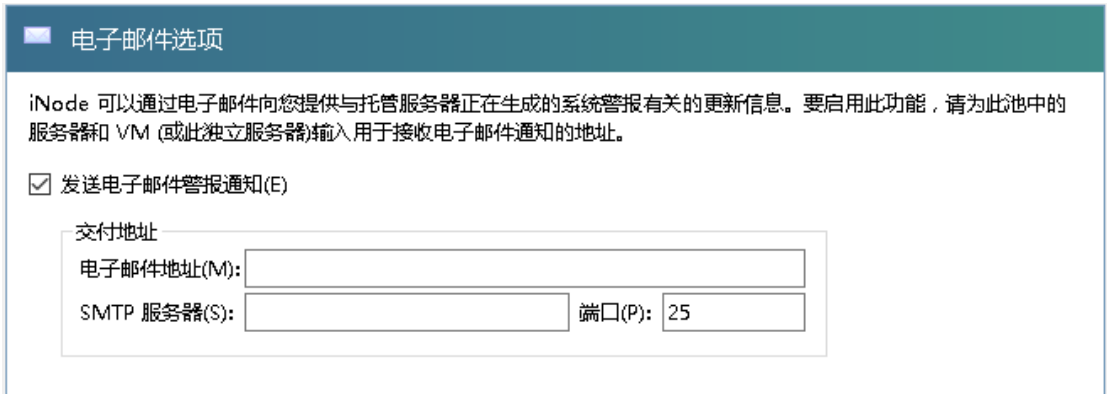


图 4.6.3-2

4.7 vApp

vApp 是一个或多个相关虚拟机 (VM) 的逻辑组，在发生灾难时，这些虚拟机可以作为单个实体来启动。当 vApp 启动后，其中包含的 VM 将按照用户预定义的顺序启动，使相互依赖的 VM 自动排列。在整个服务需要重新启动时（例如在软件更新后），管理员不再需要手动设置相关 VM 的启动顺序。vApp 中的 VM 不必位于同一个主机上，而是按照正常的规

则在池内分布。在灾难恢复的情况下，vApp 功能尤其有用。此时，管理员可以选择将位于同一个存储库中或者与同一个服务级别协议(SLA)相关的所有 VM 组合到一起。

使用 iCenter 的管理 vApp 对话框，可以在选定池内创建、删除、修改、启动、关闭、导入和导出 vApp。当您在列表中选择某个 vApp 时，其中包含的 VM 会列在右侧的详细信息窗格中。如图 4.7-1

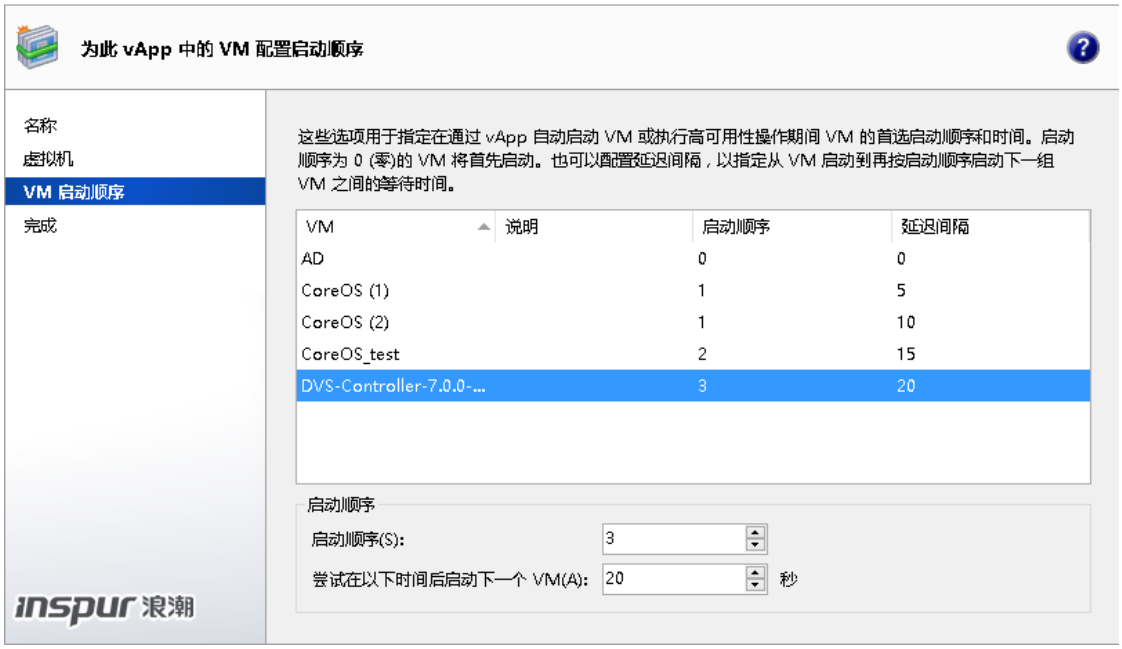


图 4.7-1

4.8 灾备

InCloud Sphere 灾难恢复 (DR) 功能旨在允许您从禁用或破坏整个池或站点的灾难性硬件故障中恢复虚拟机 (VM) 和 vApp。

4.8.1 DR 结构

使用 InCloud Sphere DR，需要在主站点和辅助站点均设置适当的 DR 基础结构：

Ø 用于池元数据和 VM 所用虚拟磁盘的存储都必须从主（生产）环境复制到备份环境。存储复制（例如，使用镜像）最好通过存储解决方案进行处理，并且因设备而异。

Ø 当 VM 和 vApp 恢复到 DR 站点上的池中并且启动并运行后，还必须复制包含 DR 池元数据和虚拟磁盘的 SR，以便恢复后的 VM 和 vApp 在主站点重新联机后立即还原到主站点（故障恢复）。

Ø DR 站点的硬件基础结构不必与主站点的硬件基础结构一致，但是 InCloud Sphere 环境必须与主站点具有相同的版本和修补程序级别，而且应当在目标池中配置足够的资源，以便重新创建和启动所有故障转移的 VM。

4.8.2 DR 工作原理

将恢复业务关键型 VM 和 vApp 所需的全部信息存储在存储库 (SR) 中，然后将存储库从主（生产）环境复制到备份环境。当主站点上受到保护的池出现故障时，可以从复制的 存储恢复该池中的 VM 和 vApp 并在辅助 (DR) 站点上重新创建，从而最大限度地减少对应用程序或造成的停机时间。如图 4.8.2-1

图 4.8.2-1

在 DR 环境中，会根据池元数据（有关池中所有 VM 和 vApp 的配置信息）在辅助 (DR) 站点上重新创建 VM。每个 VM 的元数据都包含其名称、说明、通用唯一标识符 (UUID)、内存、虚拟 CPU、网络连接配置 和存储配置。元数据还包含 VM 启动选项（启动顺序、延迟间隔和高可用性重新启动优先级），当在高可用性或 DR 环境中重新启动 VM 时将使用这些选项。例如，在灾难恢复期间恢复 VM 时，vApp 中的 VM 将 按照 VM 元数据中指定的顺序，以指定的延迟间隔在 DR 池中重新启动。

4.8.3 DR 故障转移

InCloud Sphere 灾难恢复中可以支持三种模式，分别是“故障转移”、“故障恢复”、“测试故障转移”。若当前池出现故障可根据此功能转移到灾难恢复的站点上的池，可根据 iCenter“灾难恢复向导”配置。如图 4.8.3-1

图 4.8.3-1

4.8.4 备份机制

浪潮建议无需再InCloud Sphere 旗舰版主机上安装任何其他备份软件或启用其他备份软件相关服务。

对于 VM，最佳方法是将其视为标准物理服务器，在上面安装备份代理软件。目前Windows VM,已经测试了CA BrightStor ARCserver Backup、Symantec NetBackup 和 Backup Exec。

备注：不要在控制域（InCloud Sphere dom0）中创建备份

浪潮建议按照 InCloud Sphere 管理员手册进行备份的操作，或从可能的服务器/软件中恢复

备份方法

InCloud Sphere 旗舰版备份的执行过程需要在 CLI 模式下操作，具体的命令请参考（InCloud Sphere 管理员手册）。

- 备份池元数据，池（InCloud Sphere 资源池概念，类似于主机集群）
- 备份主机配置和软件，在备份过程中可能会创建比较大的备份文件，因此需要一段过程
- 备份 VM 元数据，在备份 VM 时确保 VM 处于脱机状态，将用于备份 VM 的所有数据，因此需要消耗一段时间

还原方法

- 从特定的备份还原InCloud Sphere 主机，确保还原的主机已启动并可通过 CLI 方式连接，具体还原命令请参考《InCloud Sphere 旗舰版管理员手册》；此处还原命令仅对压缩的备份文件进行解压缩并将其还原到正常形式，但它写入其他分区并且不会覆盖当前版本的文件系统
- 使用已还原版本的根文件系统，需使用 InCloud Sphere 安装 CD 重新引导 InCloud Sphere 主机，再次选择还原选项

4.9 容器

4.9.1 Docker 介绍

Docker 是一个开源的应用容器引擎，开发者可以打包应用以及依赖包到一个可移植的容器中，容器是完全使用沙箱机制，相互之间不会有任何借口。

Docker 使用客户端-服务器（C/S）架构模式，使用远程 API 管理和创建 Docker 容器。Docker 容器通过Docker 镜像来创建。容器与镜像的关系类似于面向对象编程中的对象与类。

4.9.2 InCloud Sphere 旗舰版和 Docker

InCloud Sphere 最佳容器支撑平台：

- 提供可见的管理控制平台
- 保障容器的安全、可信任
- 具备整合容器的管理工具框架

Docker 和 InCloud Sphere 旗舰版为运行应用程序提供基础设施平台

- InCloud Sphere 旗舰版可查看哪些虚拟机运行 docker 应用、在虚拟机上查看 docker 应用的运行状态、以及直接控制容器；图 4.9.2-1

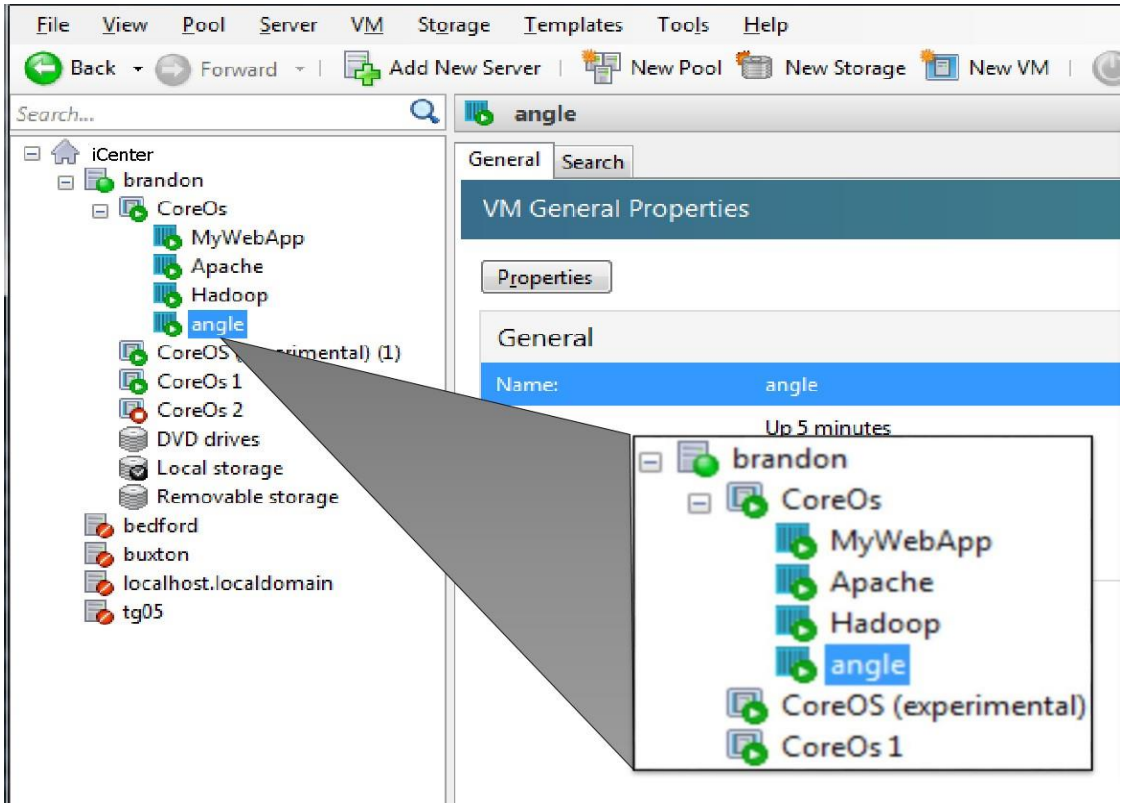


图 4.9.2-1

- 查看 docker 和容器特定的配置和诊断信息，图 4.9.2-2

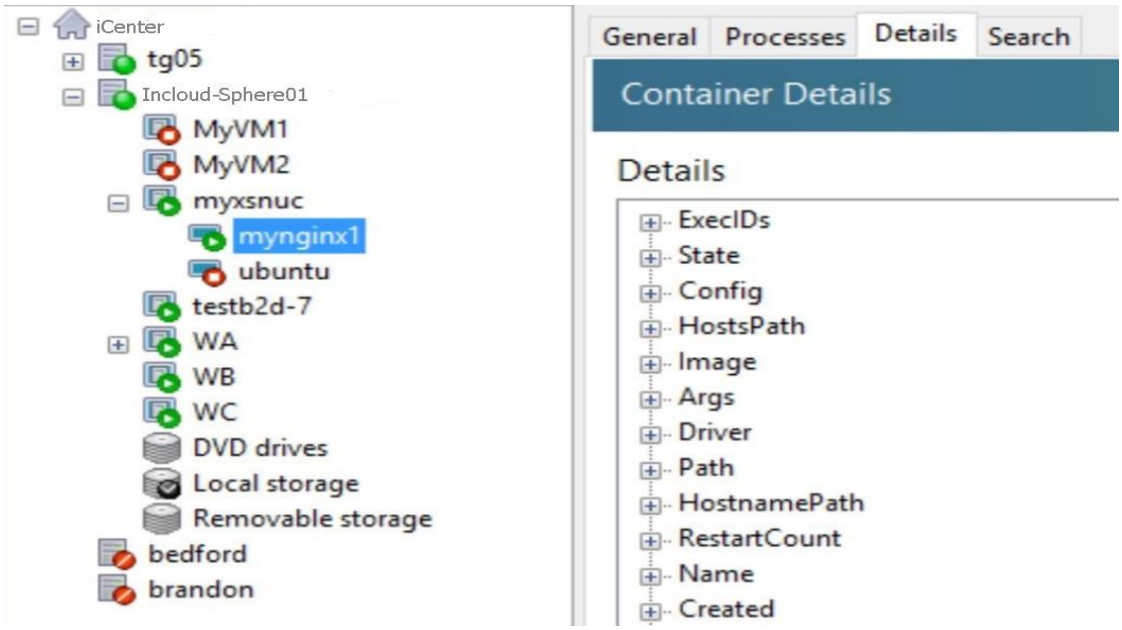


图 4.9.2-2

- 查看正在使用的资源，图 4.9.2-3

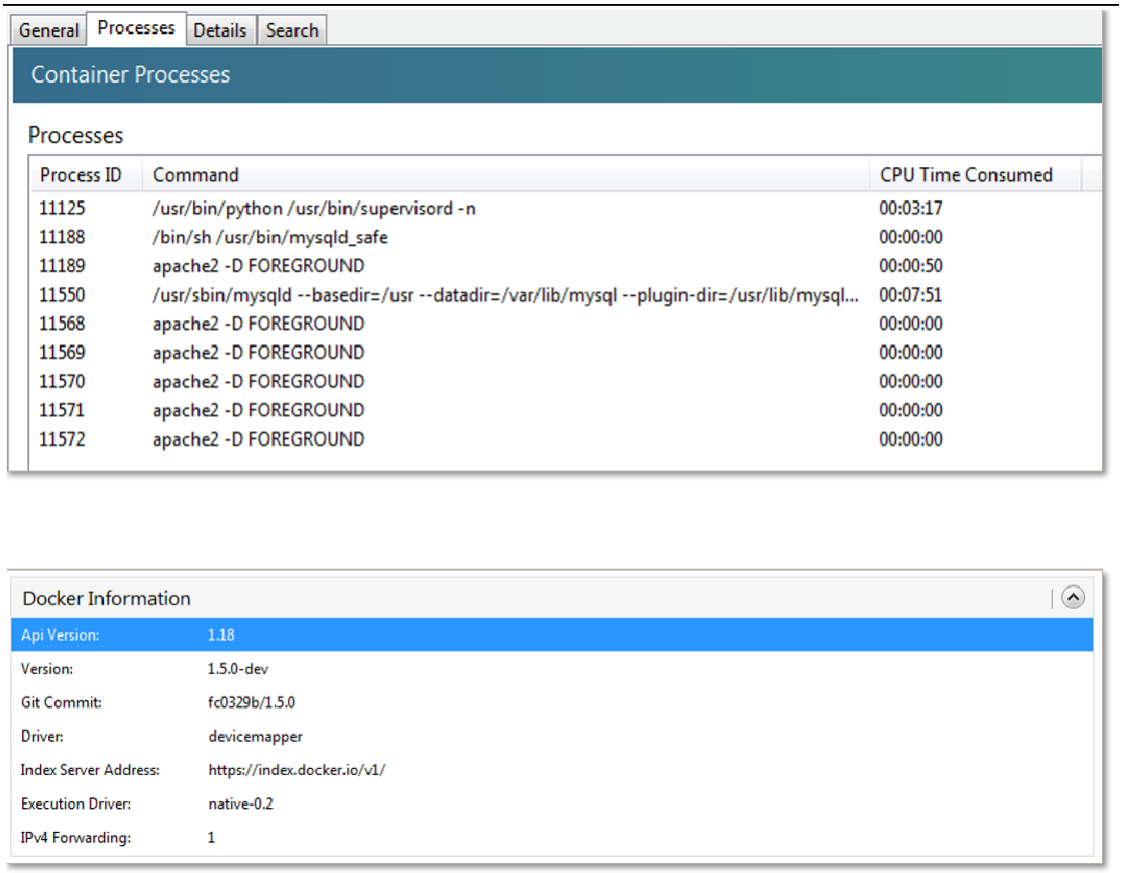


图 4.9.2-3

- 跟进异常的容器，并进行隔离和终止操作，图 4.9.2-4

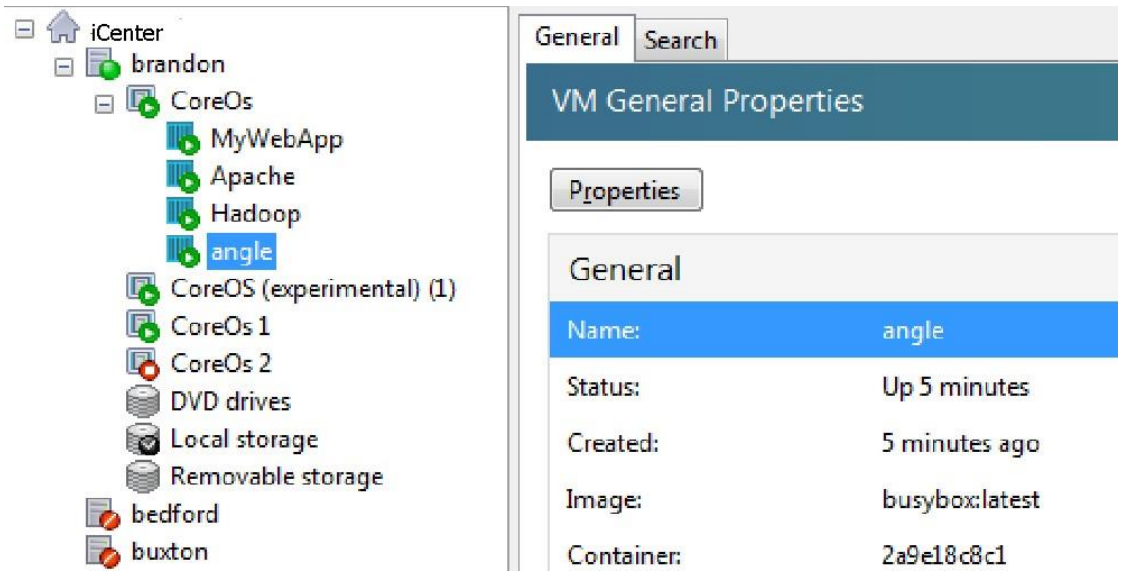


图 4.9.2-4

4.9.3 InCloud Sphere 提供 Docker 支持优势

- 提供简单和低风险的 Docker-style 应用部署

- 提高安全性、可靠性和可服务性 Docker 应用部署
- 无缝或过度开发工作流的生产部署
- 利用现有的数据中心或云资源平台部署 Docker 应用程序，无需额外的配置
- InCloud Sphere 旗舰版解决方案允许管理员以满足他们的目标在安全、合规、可用性和成本管理，同时保留容器易用性、可伸缩性和云可移植性

5 第五章 InCloud Sphere 自动化能力

通过阅读本章，您可以了解到：

- InCloud Sphere 旗舰版系统实现自动安装方法和原理；
- InCloud Sphere 旗舰版系统实现自动升级的方法和原理。

5.1 自动化安装

InCloud Sphere 旗舰版虚拟化产品的优势体现在自动化能力上，InCloud Sphere 旗舰版不仅体现在 CPU、内存等资源的自动化调度上，InCloud Sphere 旗舰版虚拟化平台的部署和升级也拥有完全的自动化能力。

5.1.1 自动化部署架构

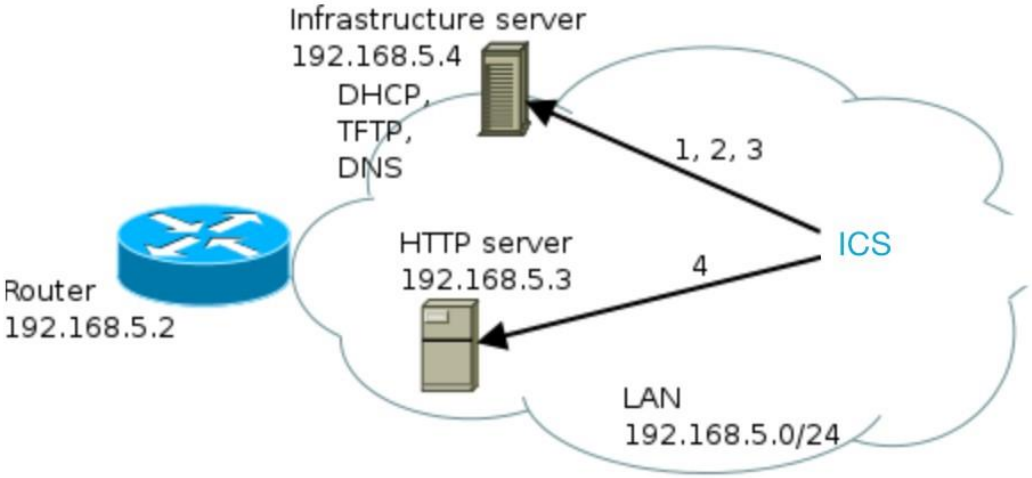


图 5.1.1-1 自动化部署架构图

5.1.2 自动化部署条件

- 部署环境网络必须联通，网络环境中必须存在 DHCP Server
- 根据获取镜像文件的方式不同（TFTP、HTTP、NFS）配置不同的镜像服务器
- 根据不同的需求配置应答文件 answer.xml

5.1.3 自动化部署过程

- Ø 服务器开机加电，BIOS 引导，PXE 启动
- Ø 从 DHCP Server 获取 IP 地址和 TFTP 的 IP 地址
- Ø 从 TFTP Server 获取 bootstrap 文件
- Ø 根据 pxelinux.cfg/default 配置文件启动系统
- Ø 加载 kernel 和 install.img
- Ø TFTP、HTTP、NFS 等方式获取安装文件（具体方式根据具体情况），根据 pxelinux.cfg 设置应答文件，实现自动化安装

5.1.4 应答文件

在我们手动安装操作系统时，需要回答各种提示问题，而针对 InCloud Sphere 旗舰版，这种回答完全有章可循，我们可以在安装操作系统时将其中出现的各种问题的应答内容提前写好（应答文件），使整个操作系统的安装就可以自动完成。

5.2 自动化更新

5.2.1 iCenter 自动检查可用更新

InCloud Sphere 旗舰版的 iCenter 在连接到 Internet 的情况下，不但能自动检查 client 端的的更新，而且能检查 iNode 节点的更新。

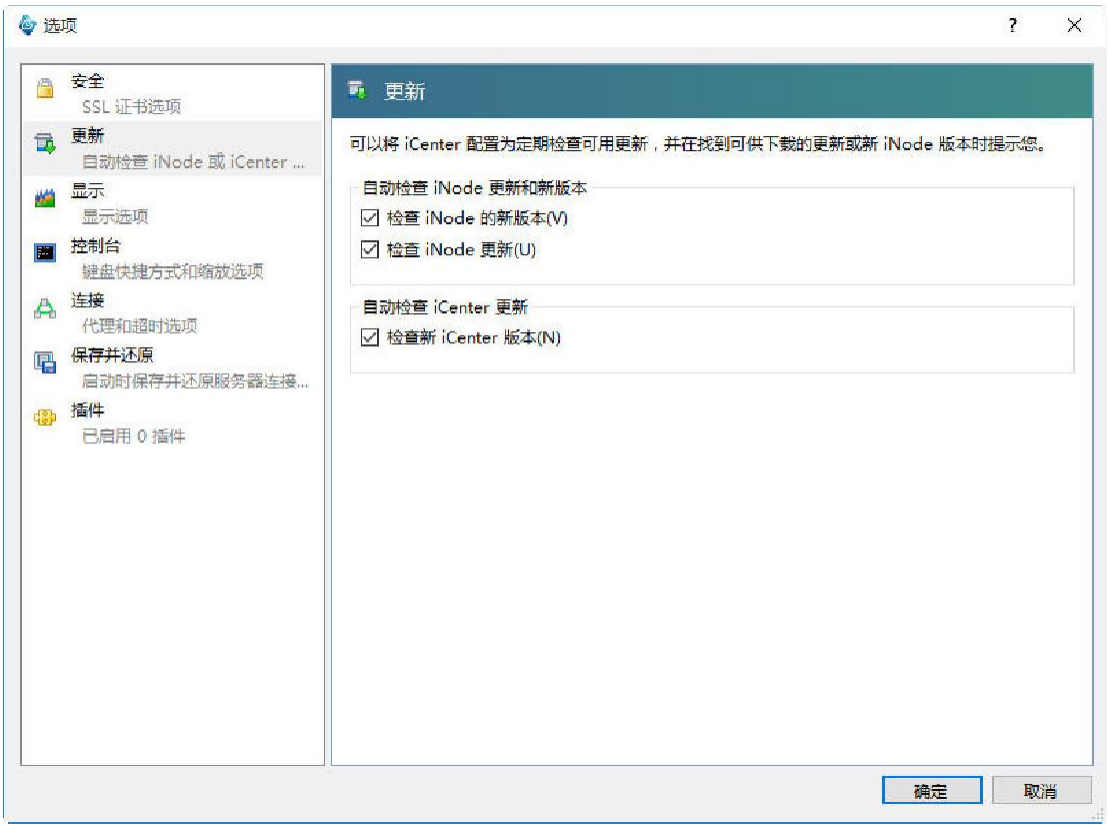


图 5.2.1-1

5.2.2 Hotfix 自动更新

Hotfix 是浪潮 InCloud Sphere 旗舰版虚拟化软件对补丁包的定义名称，浪潮会不定期的发布的 Hotfix 来修复一些 BUG 和 ISSUE，更新 Hotfix 对于管理员来说也是日常工作之一，自动化的升级 Hotfix 不仅可以减少运维管理员的工作量，而且可以更快捷安全的升级补丁。

InCloud Sphere 旗舰版支持自动下载 Hotfix 更新和手动加载 Hotfix 更新两种方式，且对于升级的 Hotfix 都会自动的执行预检查操作，有效的避免了升级 Hotfix 所产生的意外情况发生。

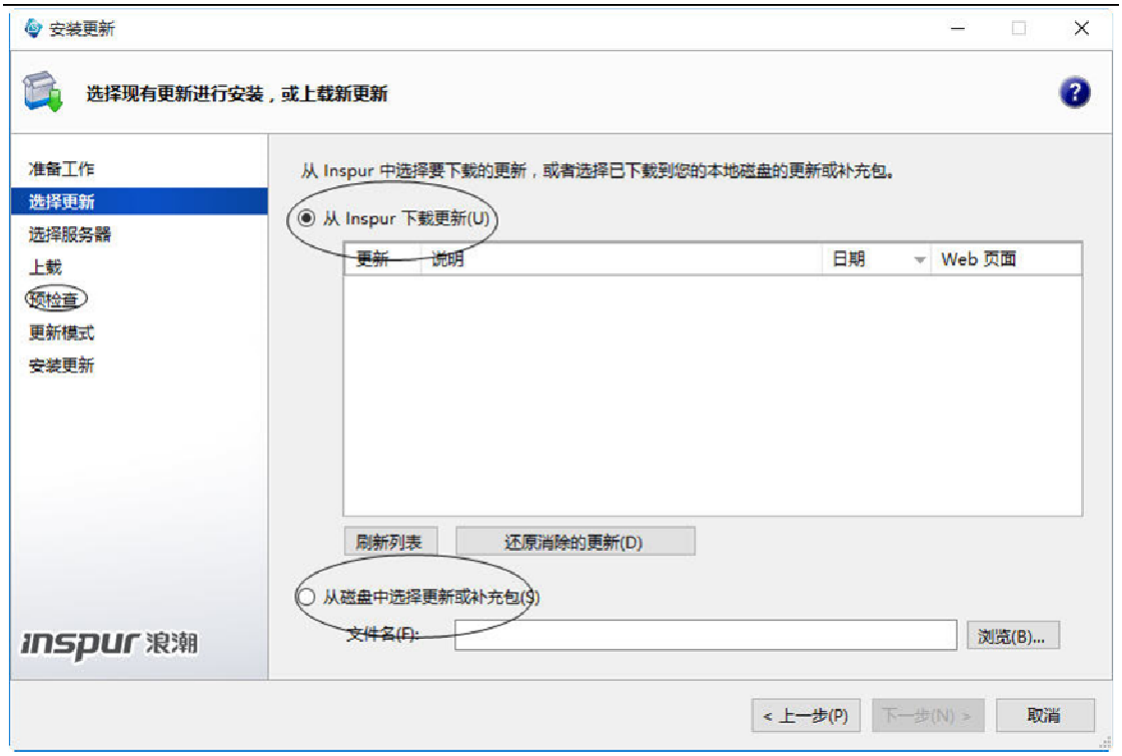


图 5.2.2-1

5.2.3 InCloud Sphere Tools 自动更新

InCloud Sphere Tools 是为了增强 InCloud Sphere 旗舰版虚拟化平台上的虚拟机的性能以及拥有完全被支持的管理功能，Tools 主要包含 PV 驱动和 管理代理。

针对 windows 虚拟机，InCloud Sphere 旗舰版提供自动化的 Microsoft Windows VM 的驱动程序更新功能：

- InCloud Sphere 旗舰版 将基于 Windows VM 的 I/O 驱动程序更新过程转移到 Microsoft Windows 更新服务。
- 在未安装 InCloud Sphere Tools 的 VM 上，这将安装一个同样是最新版本的精简版代理与半虚拟化驱动程序。
- 提供一个升级工具，用于将现有 VM 一次性批量更新到最新的驱动程序。

  在大规模的环境下，这种方式可以大大减少运维人员的工作量，使升级变得方便快捷。

5.2.4 网络升级

池滚动升级是针对 InCloud Sphere 旗舰版的大版本升级而设计的功能，使用此功能可以简化每次升级的繁琐过程，简化了运维管理员的操作内容，使升级变得便捷高效。

池滚动升级支持自动模式和手动模式，且两种方式都会自动的执行升级预检查操作，有 效的避免了升级过程中所产生的意外情况发生。

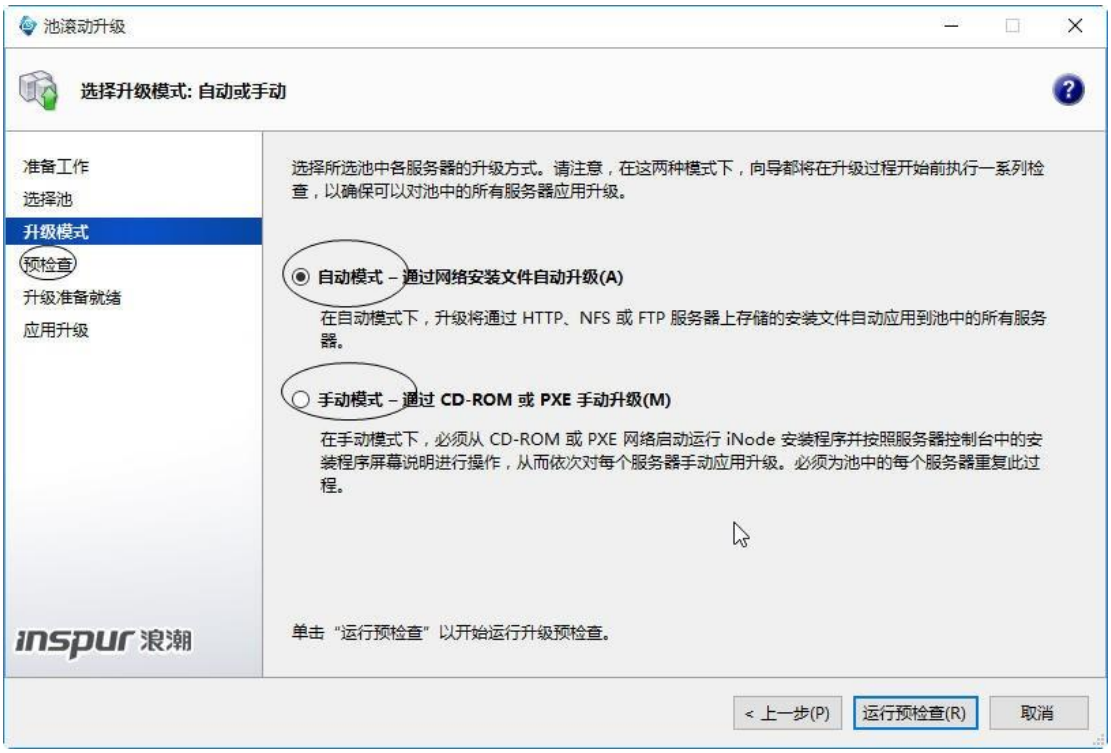


图 5.2.4-1

6 第六章 InCloud Sphere 开放性和安全性

InCloud Sphere 旗舰版虚拟化软件设计的初衷是提供稳定灵活的底层基础设施抽象化。以强大的Xen Project Hypervisor 为基础，提供高效的 Windows 和 Linux 虚拟机管理， 并提供具有极高成本效益的应用程序与服务器整合平台，提供开放的安全的虚拟化计算平台，为云计算提供良好的基础。

InCloud Sphere 旗舰版作为云平台的组件，不仅仅使用了完全自主研发的软件和硬件，也广泛的支持通用型工业标准服务器（x86 架构服务器）和业界主流厂商的通用设备（网络交换机、存储等）。

InCloud Sphere 旗舰版提供全功能的 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。

支持 Direct Inspect API，可以与第三方安全厂商集成安全产品，实现无代理防病毒功能。除此之外，还提供基于 hypervisor 层面的安全加固产品 SSR，通过对 hypervisor 的安全加固大大提高了底层系统的安全性可靠性。

6.1 XAPI

6.1.1 XAPI 介绍

XAPI 是 InCloud Sphere 旗舰版中的一组管理接口的统称，是 InCloud Sphere 旗舰版管理的核心，由一系列的 toolstack 组成。

XAPI 主要提供 iCenter 以及 Pool 中各主机通信的接口。iCenter 通过 XAPI 来读取 InCloud Sphere 旗舰版的配置、管理、License 的管理、数据库的维护等等，同时也包括如存储（SR）、虚机、虚拟网卡、HA 等等所有的功能控制。而 Pool 中的所有 InCloud Sphere 旗舰版的操作请求也是通过 XAPI 传递给 Dom0，同时在池中的所有主机中间通信，例如：Pool 中数据库（配置数据库，由 InCloud Sphere 旗舰版维护的一个小型数据库）会通过 XAPI 在所有的主机之间同步，以便在 Master 服务器宕机以后，其他机器能够正确而迅速的取代 Master，并维持 Pool 的功能和服务。

简而言之，XAPI 就是个和底层通信的中间层、接口层。

6.1.2 XAPI 功能

Ø XAPI 主要提供 iCenter 以及 pool 中各主机通信的接口；

Ø iCenter 通过 XAPI 来读取 InCloud Sphere 旗舰版的配置、管理、License 的管理、数据库的维护等等，同时也包括如存储（SR）、虚机、虚拟网卡、HA 等等所有的功能控制；

Ø Pool 中的所有 InCloud Sphere 旗舰版的操作请求也是通过 XAPI 传递给 Dom0，同时在池中的所有主机中间通信；

Ø XAPI 就是个和底层通信的中间层、接口层。

6.1.3 XAPI 架构

XAPI 是基于 XML-RPC 协议进行通信的。XML-RPC 是在 Internet 上实现远程方法调用的一种规范，它利用 HTTP 作为传输协议，使用 XML 作为消息请求的传输主体。XML-RPC 将一个 XML 格式的消息体作为 HTTP POST 请求发送给服务器，该消息包括名称、运行服务的程序以及输入参数，服务器将执行结果以 XML 格式返回，其原理如图 6.1.3-1 所示：

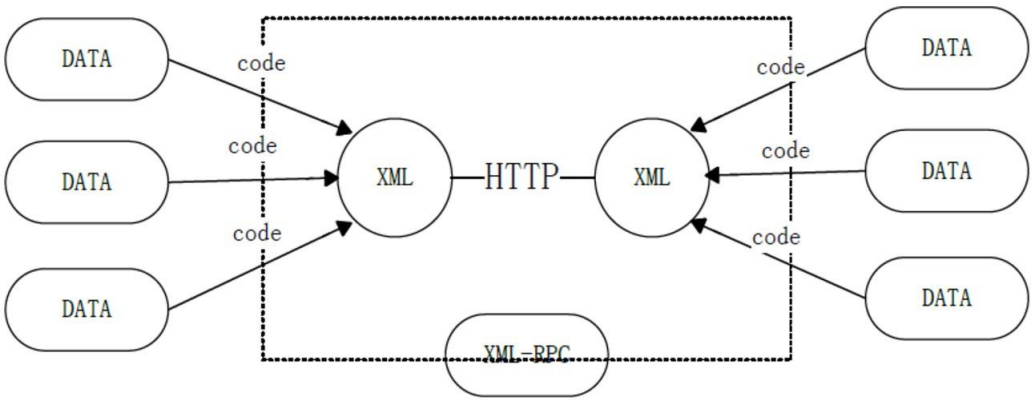


图 6.1.3-1

通过 XAPI，用户可以管理虚拟机、存储、网络、主机配置和资源池。

6.2 Introspect API

6.2.1 Introspect API介 绍

InCloud Sphere 旗舰版定期推出创新的安全功能，如代表着虚拟基础架构保护模式转变的 Direct Inspect API。这些 API 将对虚拟机的保护转移到主机，让病毒和恶意软件在来宾操作系统中无处藏身。Direct Inspect API 与安全供应商合作伙伴集成，通过这一真正的无代理防病毒解决方案提供“比物理保护更好的”病毒保护。

Direct Inspect API 的出现是革命性的，这意味着许多安全厂商可以采用一种前所未有方式构建防病毒体系，而这些是传统的方法所无法做到。就目前而言国产虚拟化软件只有浪潮 的 InCloud Sphere 旗舰版支持。

6.2.2 虚拟机内存保护

由于对于内存的实时读取和写入需要耗费大量的资源，传统的技术注重保护虚拟机的文 件系统，Direct Inspect API 解决了这个问题，通过直接保护虚拟机的内存来防范威胁，这使得攻击可能永远触及不到文件系统层面。

保护和监控的区域示意图如图 6.2.2-1：

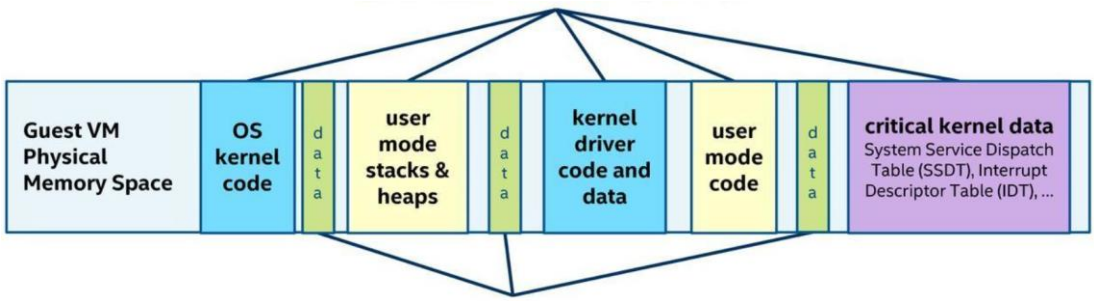


图 6.2.2-1 保护和监控的区域示意图

6.2.3 预防攻击技术

预防攻击技术不是找出病毒将其清除，而是在病毒进入系统前将其拦截，这是一个很重 要的区别，有效的避免了各种变种的恶意软件无法清除的情况发生。通过预防、拦截的技术 安全厂商可以有效的防范“not yet seen”类的高级病毒攻击。

目前浪潮正在与多家安全厂商合作，后期将会推出集成第三方杀毒软件的 InCloud Sphere 旗舰版安全解决方案。

6.2.4 虚拟机无代理保护

传统的虚拟机杀毒解决方案需要每台虚拟机上安装 agent，此方式有两种缺点：

Ø 安装 agent 耗费大量的时间

Ø 无法防御某些高级的病毒威胁，如 rootkit，可以直接攻击 agent，通过 agent 来传播。

Direct Inspect API 允许保护发生来自外部 –也就是使用虚拟机管理程序 hypervisor 提供硬件强制隔离，这意味着攻击者不能再直接攻击安全软件。

虚拟机无代理保护架构图，如图 6.2.4-1 所示：

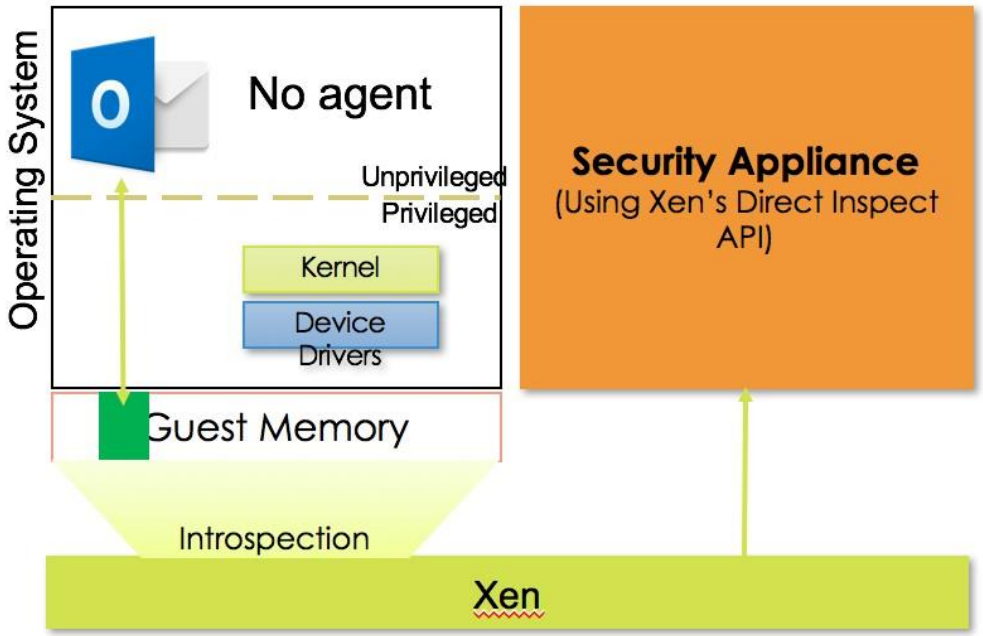


图 6.2.4-1 虚拟机无代理保护架构图

6.2.5 Direct Inspect API 防病毒架构

虚拟机某些文件收到病毒攻击后，防病毒软件通过 Direct Inspect API 直接将虚拟机的收到攻击的内存单元隔离。如图 6.2.5-1 所示

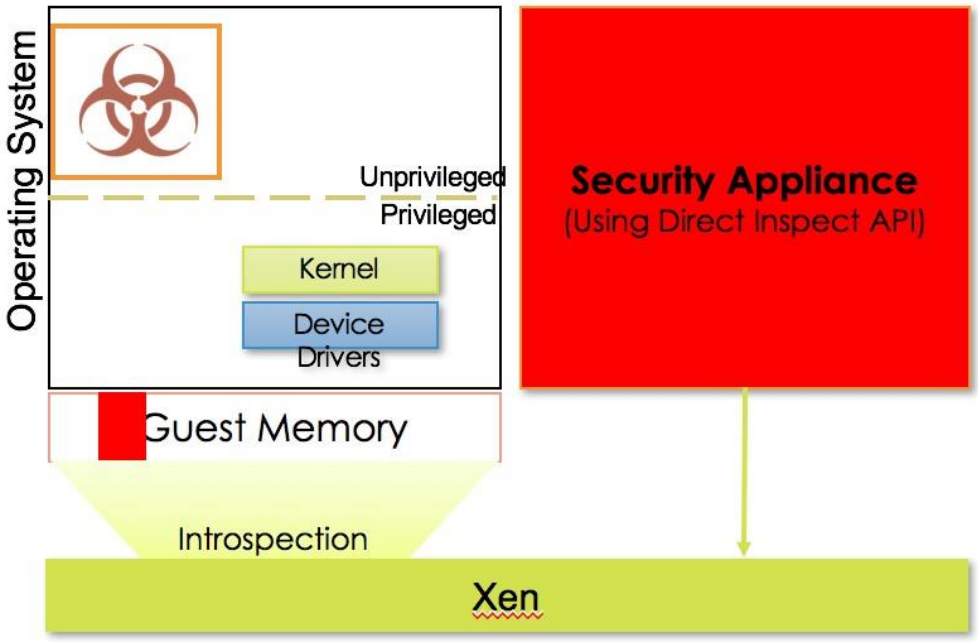


图 6.2.5-1 病毒隔离示意图

6.2.6 Direct Inspect API 防病毒的优势

Direct Inspect API 的防病毒技术弥补了现有安全技术的缺点，这是一项伟大的技术，特别是当涉及到保护您的最宝贵的 IT 资产时更能体现出它的价值。

- Ø 预防病毒攻击上升到一个新的安全层面；
- Ø 在 hypervisor 层面运行内存 introspection 可以对威胁进行直接的处理；
- Ø 从病毒的寻址和攻击技术为切入点对威胁进行处理；
- Ø 无代理，可以与现有的安全解决方案无缝隙结合；
- Ø 对于虚拟机的性能开销极小可以处理多种的高级病毒程序。

6.3 PlugIn

6.3.1 PlugIn 介绍

InCloud Sphere 旗舰版为了方便扩展功能，引入了插件功能，为第三方合作厂商扩展特有功能提供支持，通过插件的方式可以增加或提高现有的 InCloud Sphere 旗舰版功能（如管理、监控、备份等），方便满足用户定制化需求，扩展原有功能。

6.3.2 PlugIn 优势

对于多数正在使用 InCloud Sphere 旗舰版的客户来说，他们的环境可能是完全的 windows 工作负载或者是完全的 Linux 工作负载，也可能是混合型的工作负载。对于不同的工作负载客户的需求也不尽相同，有些倾向于使用 powershell，而有些则倾向于是用*nix-oriented 等的自动化工具，针对这种多样化的需求，多样化的群体，InCloud Sphere 旗舰版提 供了一个非常灵活的平台，通过构建了 Plugin 的框架结构轻而易举的实现多种多样的需求。

例如我们需要一个新的 UI 来满足监控的需求，我们可以通过 InCloud Sphere 旗舰版插件功能轻松的实现的新的 UI 界面来实现相应的功能。

6.3.3 部分 PlugIn 插件列表

表 6.3-1 部分 PlugIn 插件列表

Windows-only extensions	Universal extensions	ICS Server
Establish an RDP connection to VM	Connect to VM using Putty (most useful for Linux VMs)	Connect to ICS host using Putty
Open remote Services of VM	Connect to VM using WinSCP	Connect to ICS host using WinSCP
Open remote Event Log of VM	Connect to VM using HTTP	Connect to ICS host using PowerShell
Open remote disk drives (C, D and E) of VM	Ability to Ping VM	

6.4 安全架构

InCloud Sphere 旗舰版在设计之初就考虑到安全因素，各个模块之间通信采用加密 SSL 通道，新版本将 SSL 升级至 v3，使得安全性进一步提升。

InCloud Sphere 旗舰版各模块通信架构图如图 6.4.1-1：

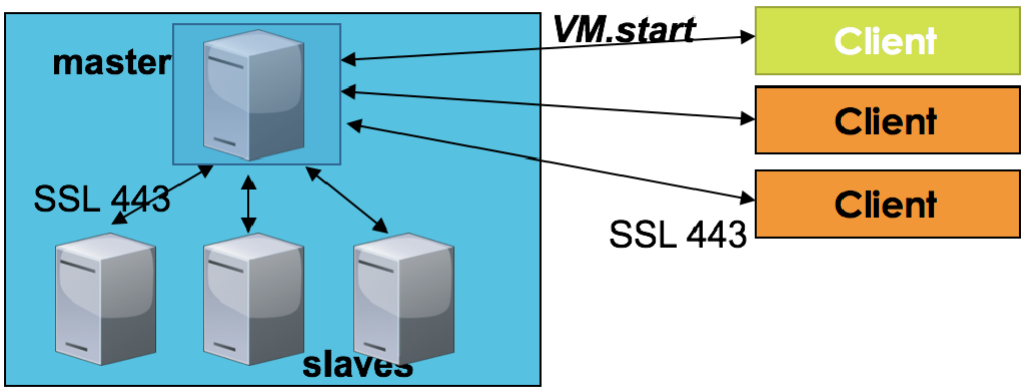


图 6.4.1-1

6.5 SSR 虚拟化安全加固

服务器虚拟化系统 InCloud Sphere 支持与主机安全加固系统系统集成，产品接口做了深度优化，全部功能已无缝兼容。

6.5.1 SSR 介绍

与传统的防火墙、入侵检测系统等基于网络防护的安全产品不同，浪潮主机安全增强系 统（system security reinforcement 简称 SSR）是基于对主机的内核级安全增强防护，当未经授权的内、外网用户通过各种手段突破了防火墙等网络安全产品进入了主机内部，浪潮操作系统安全增强系统就将成为最后也是最坚固的一道防线。

而浪潮 SSR-h 版本是专门针对虚拟化层安全加固，实现对 hypervisor 层强制访问控制等，实现虚拟化层管理员的权力进行分散，使其不再具有对系统自身安全构成威胁的能力，从而达到从根本上保障系统安全的目的。

浪潮主机安全增强系统能够稳定地工作于浪潮 InCloud Sphere 云操作系统下，提升系统的安全等级，并实现具有《计算机信息系统安全等级保护划分标准》中规定的第三级即“安全标记保护级”的网络安全产品，为用户构造一个更加安全的云操作系统平台。

浪潮（北京）信息产业有限公司的主机安全增强系统是一基于操作系统内核的安全产品。可以有效的防止来自于内部、外部网络的威胁隐患。并通过对 InCloud Sphere 云操作系统的超级用户权限进行合理分散与适度制约，从而降低超级用户权限被窃取后系统被肆意非法操作的风险。

SSR 实现了网络管理的功能，可以通过 SSR 集中管理平台对安装在 InCloud Sphere 云操作系统的 SSR 客户端进行策略分发、修改、查询、删除等功能。并可对操作时产生的日志进行记录。对于违规日志，可以进行条件查询、备份等审计活动。

本产品，SSR（集中管理模式），适用于有多台重要 InCloud Sphere 云操作系统的用户，可节约用户的成本，方便技术人员对重要服务器的管理。

6.5.2 SSR 实现原理

SSR（集中管理模式）由集中管理平台和 SSR 客户端两部分组成，SSR 客户端安装在需要保护的 InCloud Sphere 主机上面，集中管理平台安装在单独的主机上面。工作原理如图 6.5.2- 1：

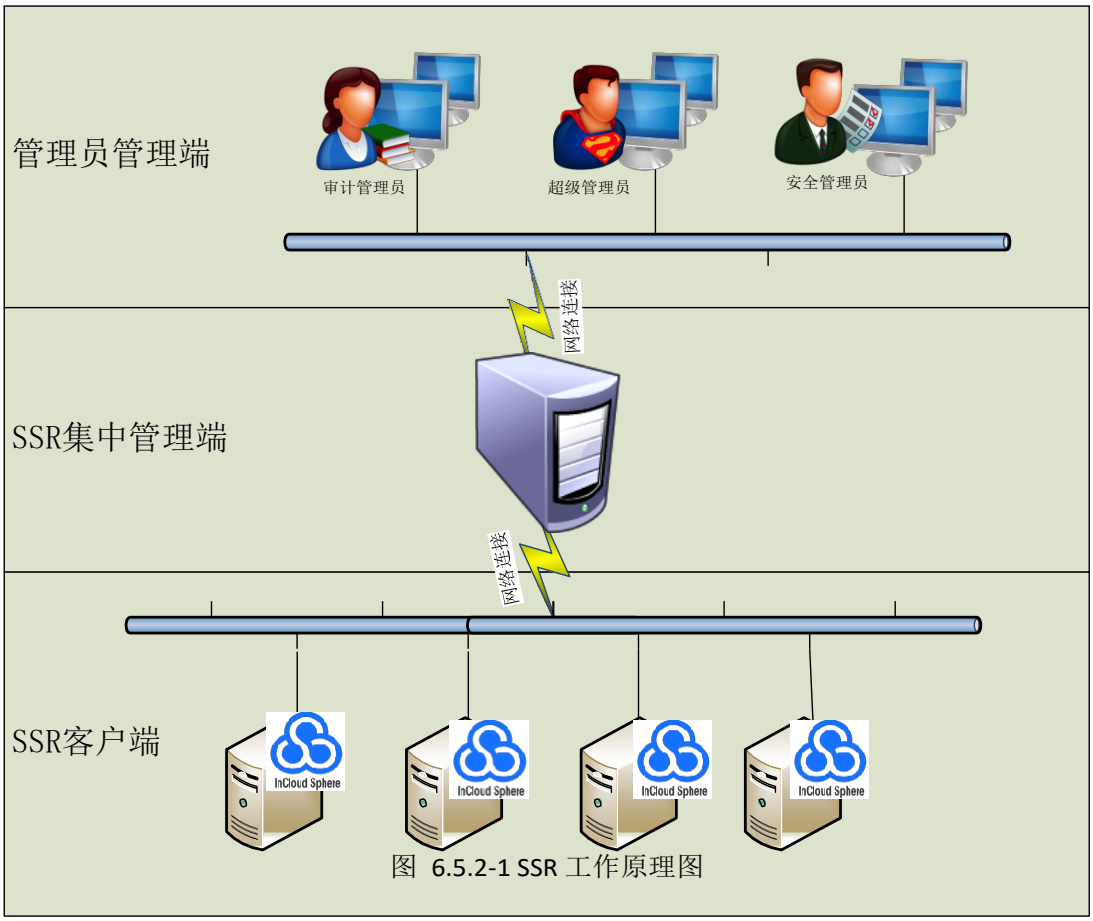


图 6.5.2-1 SSR 工作原理图

6.5.3 SSR 技术架构

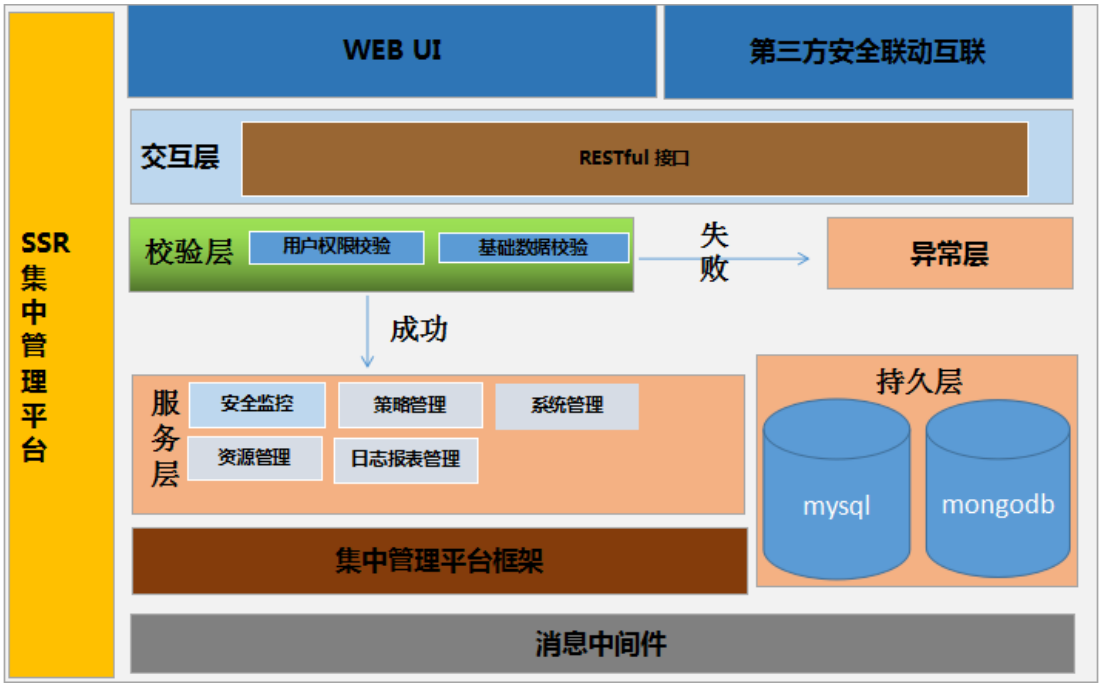


图 6.5.3-1 SSR 技术架构图

如图 6.5.3-1 所示，接口服务层分两部分，一部分为业务功能提供 Restful 接口，提供系统内部接口供业务展现层调用。另一部分为联动 API，负责向 InCloud Manager 提供安全联动功能。

校验层使用对接口数据进行数据校验和用户权限校验，校验成功会进入到不同的业务逻 辑进行处理，校验失败会统一封装异常信息包含错误描述和错误码，返回调用层。

服务层根据不同的业务分为了安全监控，策略管理，系统管理，资源管理和日志报表管理。其中安全监控可以监控客户端运行状态，授权信息和集中管理平台的运行状态；策略管理可以生成策略，编辑策略并向客户端下发策略；系统管理可以对管理平台进行系统配置； 资源管理可以管理客户端基本信息，维护客户端注册，心跳等信息，并可以进行分组；日志报表管理可以分别对日志报表进行查询，导出，配置等操作。

集中管理平台框架提供了 API 库，业务层，通信框架消息层和架构平台层。

消息总线为集中管理平台和客户端提供通信的通道，采用 RabbitMQ 作为中间件，对多个消息队列进行监听，实现管理平台和客户端的异步通信。

6.5.4 SSR 主要功能

表 6.5-1 SSR 主要功能列表

功能	说明
证书管理	安全证书、USB key
服务器管理	iNode 主机管理
用户管理	用户组、角色、权限
安全策略管理	文件的访问控制
	进程的访问控制
	网络的访问控制
状态监控	系统资源监控
	文件完整性
	安全区域监控
告警管理	告警发送
安全审计	生成审计日志
报表管理	生成报表

6.6 与 OpenStack 集 成

6.6.1 OpenStack 介 绍

OpenStack 是一个旨在为公共及私有云的建设与管理提供软件的开源项目。它的社区拥有超过 130 家企业及 1350 位开发者（现在更多），这些机构与个人都将 OpenStack 作为基础设施即服务（简称 IaaS）资源的通用前端。OpenStack 项目的首要任务是简化云的部署过程并为其带来良好的可扩展性。OpenStack 是一个云平台管理的项目，它不是一个软件。这个项目由几个主要的组件组合（Nova、Neutron、Glance、Cinder 等）起来完成一些具体的工作。

6.6.2 InCloud Sphere 旗舰版的优势

InCloud Sphere 旗舰版基于 Xen 开源内核，产品成熟稳定，功能完善，界面友好，与 OpenStack 无缝隙的集成从而提供虚拟服务器部署和计算模块。通过 InCloud Sphere 广泛的插件支持，实现与 OpenStack 无缝隙集成，实现 InCloud Sphere 与 OpenStack 共同管理您的云计算平台

InCloud Sphere 旗舰版的 Dom0 通常会包含一个工具栈(Toolstack)，其能够实现让用户完

成虚拟机的创建、删除、配置等功能。此工具栈还提供了一个访问接口，因此，其管理功能 还可以通过相应的命令行工具、图形化控制台如 OpenStack 的云计算环境来完成。通过 InCloud Sphere 旗舰版的插件支持，可以简单快速的将 OpenStack 部署在我们的虚拟化环境中。

6.6.3 与 OpenStack 集成架构图

在 OpenStack 中，Nova、Neutron、Cinder 等是运行在一个 DomU 上的，并在在 Dom0 的系统软件和 DomU 建立了一定程度的安全隔离。具体架构如图 6.6.3-1：

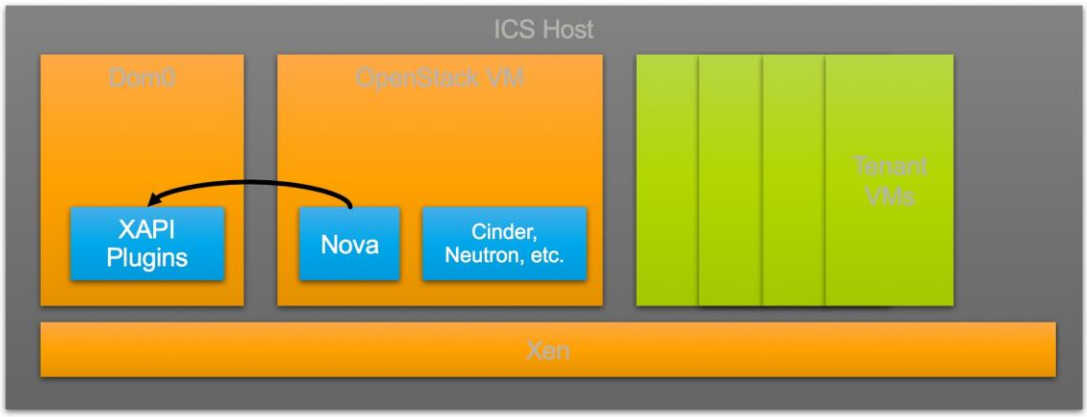


图 6.6.3-1 与 OpenStack 集成架构图

Xen VM 可分为 PV 和 HVM 两种模式，OpenStack 的 DomU 必须运行在 PV 模式下，如图 6.6.3-2 所示：

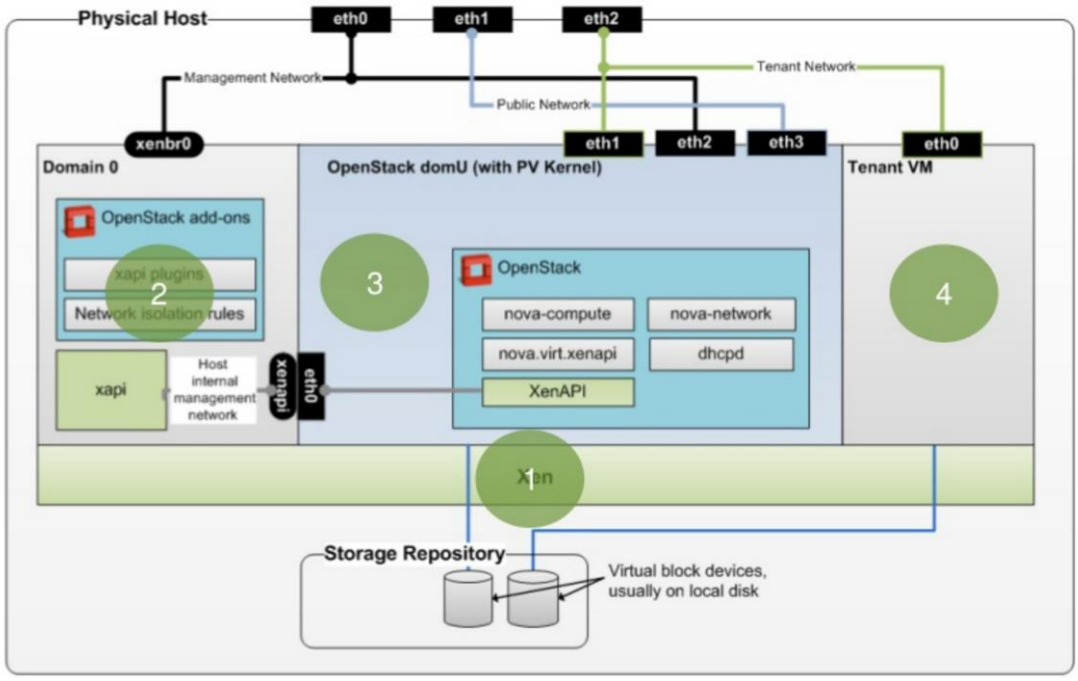


图 6.6.3-2

管理程序：Xen 内核的 InCloud Sphere 虚拟化软件。

Dom0：特权域，运行 XAPI 和 OpenStack 的插件。

OpenStack 的 DomU：计算节点运行于半虚拟化虚拟机上，在管理主机上运行。每个主机都运行一个计算节点的本地实例。Nova 使用 XAPI 的 Python 库通信，从 DomU 到 Dom0 使用主机的内部管理网络。

7 第七章 总结

InCloud Sphere 4.5 旗舰版虚拟化平台浪潮云计算解决方案的关键技术平台，主要定位企业级应用领域，采用裸金属架构的 X86 虚拟化技术，是一个基于Xen 技术自主开发的服务器虚拟化平台。通过实现对服务器物理资源的抽象，将 CPU、内存、I/O 等服务器物理资源转化为可统一管理和分配的逻辑资源，为应用提供安全隔离的虚拟机运行环境，并且充分利用硬件辅助虚拟化技术，使其具有高性能、可用性和安全性特点，实现更低的运营成本、

更高的自动化管理水平和更快速的业务响应速度。结合浪潮在高端服务器、海量存储、云操 作系统、信息安全技术等方面的先进技术，为客户打造领先的云基础架构，在政务云、行业 云、企业云等方面，提供全面的支撑和解决方案。

8 第八章 缩略语

表 8-1 缩略语清单

中文全称	英文缩写	英文全称
访问控制列表	ACL	Access Control List
活动目录	AD	Active Directory
应用软件编程接口	API	Application Programming Interface
地址解析协议	ARP	Address Resolution Protocol
主板管理控制器	BMC	Baseboard Management Controller
询问握手认证协议	CHAP	Challenge Handshake Authentication Protocol
通用互联网文件系统	CIFS	Common Internet File System
命令行界面	CLI	Command Line Interface
公共对象请求代理体系结构	CORBA	Common Object Request Broker Architecture
中央处理器	CPU	Central Processing Unit
动态主机配置协议	DHCP	Dynamic Host Configuration Protocol
动态内存控制	DMC	Dynamic Memory Controller
动态内存范围	DMR	Dynamic Memory Range
域名系统	DNS	Domain Name System
容灾	DR	Disaster Recovery
戴尔远程访问控制器	DRAC	Dell Remote Access Controller
动态资源调度	DRS	Dynamic Resource Scheduler
分布式虚拟交换机	DVS	Distributed Virtual Switch
分布式虚拟交换机管理	DVSM	Distrubuted Vitural Switch Management
差错检查及纠正	ECC	Error Checking and Correction
扩展页表	EPT	Extended Page Table
最终用户许可协议	EULA	End User License Agreement
弹性虚拟交换机	EVS	Elastic Virtual Switch
光纤通道	FC	Fiber Channel
以太网光纤通道	FCoE	Fibre Channel over Ethernet
光纤通道存储区域网络	FCSAN	Fiber Channel Storage Area Network
完全合格域名	FQDN	Fully Qualified Domain Name
文件传输协议	FTP	File Transfer Protocol
客户机物理地址	GPA	Guest Physical Address
图形处理器	GPU	Graphics Processing Unit
客户机虚拟地址	GVA	Guest Virtual Address
高可靠性	HA	High Avialability
主机总线适配器	HBA	Host Bus Adapter

主机物理地址	HPA	Host Physical Address
HP 远程管理端口	HP-iLO	Hewlett-Packard Integrated Lights-Out
超文本传输协议	HTTP	Hypertext Transfer Protocol
硬件虚拟机	HVM	Hardware Virtual Machine
基础设施即服务	IaaS	Infrastructure as a Service
互联网数据中心	IDC	Internet Data Center
输入输出	I/O	Input and Output
每秒读写（I/O）操作次数	IOPS	Input/Output Operations Per Second
网络互连协议	IP	Internet Protocol
智能平台管理接口	IPMI	Intelligent Platform Management Interface
存储区域网络	IPSAN	IP Storage Area Network IP
互联网传输协议安全性	IPSec	Internet Protocol Security
iSCSI 限定名称	IQN	iSCSI Qualified Name
指令集	ISA	Instruction Set Architecture
因特网小型计算机系统接口	iSCSI	Internet Small Computer Systems Interface
独立软件供应商	ISV	Independent Software Vendor
Java 归档文件	JAR	Java Archive
Java 商用集成	JBI	Java Business Integration
Java 开发工具包	JDK	Java Development Kit
Java 消息服务	JMS	Java Message Service
链路汇聚控制协议	LACP	Link Aggregation Control Protocol
局域网	LAN	Local Area Network
轻量目录访问协议	LDAP	Lightweight Directory Access Protocol
逻辑单元编号	LUN	Logical Unit Number
逻辑卷管理器	LVM	Logical Volume Manager
介质访问控制层	MAC	Media Access Control
城域网	MAN	Metropolitan Area Network
存储器管理单元	MMU	Memory Management Unit
最大传输单位	MTU	Maximum Transmission Unit
网络端口地址转换	NAPT	Network Address Port Translation
网络附属存储	NAS	Network Attached Storage
网络地址转换	NAT	Network Address Translation
网络文件系统	NFS	Network File System
网卡	NIC	Network Interface Controller
网络时间协议	NTP	Network Time Protocol
操作系统	OS	Operating System
开放虚拟化设备	OVA	Open Virtualization Appliance
开放虚拟化格式	OVF	Open Virtualization Format
平台即服务	PaaS	Platform as a Service

物理块设备	PBD	Physical Block Device
预启动执行环境	PXE	Preboot Execute Environment
独立冗余磁盘阵列	RAID	Redundant Array of Independent Disks
服务质量	Qos	Quality of Service
内存	RAM	Random Access Memory
基于角色的访问控制	RBAC	Role-Based Access Control
远程桌面协议	RDP	Remote Desktop Protocol
表示状态转移	REST	Representational State Transfer
数据恢复点	RPO	Recovery Point Objective
业务恢复时间	RTO	Recovery Time Objective
半虚拟化	PV	Paravirtualization
软件即服务	SaaS	Software as a Service
存储区域网络	SAN	Storage Area Network
小型计算机系统接口	SCSI	Small Computer Systems Interface
服务器信息块	SMB	Server Message Block
源地址转换	SNAT	Source Network Address Translation
简单对象访问协议	SOAP	Simple Object Access Protocol
存储库	SR	Storage Repositories
安全套接层	SSL	Secure Sockets Layer
传输控制协议	TCP	Transmission Control Protocol
磁带库	TL	Tape Library
用户数据报协议	UDP	User Datagram Protocol
统一分布式存储	UDS	Universal Distributed Storage
用户界面	UI	User Interface
统一资源定位器	URL	Uniform Resource Locator
通用串行总线	USB	Universal Serial Bus
统一虚拟化平台	UVP	Unified Virtualization Platform
虚拟块设备	VBD	Virtual Block Device
虚拟数据中心	VDC	Virtual Data Center
虚拟桌面基础结构	VDI	Virtual Desktop Infrastructure
存储磁盘映像	VDI	Virtual Disk Image
虚拟以太网桥	VEB	Virtual Ethernet Bridge
虚拟以太网端口聚合	VEPA	Virtual Ethernet Port Aggregator
虚拟硬盘	VHD	Virtual Hard Disk
虚拟接口	VIF	Virtual Interface
虚拟智能存储	VIS	Virtual Intelligent Storage
虚拟局域网	VLAN	Virtual Local Area Network
虚拟机	VM	Virtuak Machine
虚拟机设备队列	VMDQ	Virtual Machine Device Queue
虚拟机硬盘格式	VMDK	VMWare Virtual Machine Disk Format
虚拟机文件系统	VMFS	VMWare Virtual Machine File System
虚拟机管理器	VMM	Virtual Machine Monitor
虚拟私有云	VPC	Virtual Private Cloud

虚拟私有网络	VPN	Virtual Private Network
虚拟交换机端口	VSP	Virtual Switch Port
卷影复制服务	VSS	Volume Shadow Copy Service
标准虚拟交换机	vSS	Standard Virtual Switch
虚拟交换机	vSwitch	Vitural Switch
虚拟磁带库	VTL	Virtual Tape Library
广域网	WAN	Wide Area Network
动态工作负载均衡	WLB	Workload Blancing
LAN 唤醒功能	WOL	Wake On Lan
扩展标记语言	XML	Extensible Markup Language
Xen 虚拟设备	XVA	Xen Virtual Appliance