



构建 VMware 混合云平台

通过采用英特尔的硬件组件、VMware 的混合云软件和 VMware Cloud on AWS，企业可加快工作负载部署，简化管理并强化自身竞争优势

英特尔数据平台事业部

作者

Patryk Wolsza

云解决方案架构师 vExpert

Karol Brejna

高级架构师

Marcin Hoffmann

云解决方案工程师

Lukasz Sitkiewicz

软件工程师

其他贡献者

Marek Małczak

云解决方案工程师

Ewelina Kamyszek

技术部门本科实习生

Piotr Grabuszynski

软件工程师

VMware

作者

Rick Walsworth

VMware Cloud 高级产品线营销经理

Enrique Corro

CTO 办公室数据科学主管工程师

Christopher Martin

VMware Cloud 云解决方案工程师

执行概要

为了提高自身竞争力并跟上新技术的发展，所有企业都必须克服关键挑战，以加快产品开发，减少停机时间和维护开销，并以更低的成本获得更有力的竞争优势。如今，技术是一切新变化的核心。传统的应用和服务托管方法已无法实现业务所需的创新速度。

应对这一挑战需要加快整体软硬件的置备、部署和维护生命周期的发展，同时加速应用的开发、测试和交付。最终用户自助服务解决方案预计将进一步缩短上市时间。针对这些新的市场需求，VMware Cloud Foundation 和英特尔提供了一个易部署、易管理的混合云平台，用于虚拟机 (VM) 管理和容器编排。该解决方案提供了跨私有云和公有云的基础设施和运维方式。其中，部署在本地关键基础设施中所使用的英特尔® 硬件组件带来了出色性能和可靠性。

本参考架构将涵盖以下问题：

- 如何准备和部署一个连接到 VMware Cloud on AWS 的 VMware Cloud Foundation 环境？
- 如何利用相关的英特尔® 技术，如英特尔® 傲腾™ 持久内存？

由于“云计算”一词现在经常与虚拟机和容器化的使用联系在一起，本参考架构对其多种应用进行了说明，包括主流的基于虚拟机的数据库解决方案，如 Oracle 和 Microsoft SQL，以及基于容器的数据分析和人工智能 (AI) 工作负载。这些用例突出了解决方案的灵活性和整体性能。

本参考架构的目标受众包括系统管理员和系统架构师。我们假设受众对虚拟化技术、网络 (VMware NSX-T) 和 Kubernetes 有一定经验，对机器学习和人工智能的核心概念也有一定的了解。

目录	物料清单	15
执行概要	部署模块	17
简介：为什么要为您的混合云选择 VMware Cloud Foundation?	VMware Cloud Foundation	17
2	VMware vSAN	17
解决方案概述	VMware NSX	18
3	Tanzu Kubernetes Grid Integrated	18
技术介绍	集成 TKGi、VMware Cloud Foundation、NSX-T 和 vSAN	19
3	VMware Cloud on AWS	20
英特尔® 硬件	环境配置和部署	20
云基础设施	在 VMware ESXi 上安装和配置英特尔® 傲腾™ 持久内存模组	20
数据仓库构建模块	为英特尔® 傲腾™ 持久内存准备虚拟机	20
数据分析和人工智能构建模块	环境置备	21
自助应用目录	软硬件要求	21
经平台验证的数据仓库工作负载性能	VMware Cloud Foundation 管理域部署	21
9	VMware Cloud Foundation 工作负载域部署	22
Linux 上的 Microsoft SQL Server 容器	在新创建的工作负载域中添加 NSX-T Edge 主机	23
独立虚拟机上的 Microsoft SQL Server - 概述	Tanzu Kubernetes Grid Integrated 部署	23
9	VMware Cloud on AWS 配置	23
App Direct 模式下的 Microsoft SQL Server - 配置和基准测试结果	总结	25
10	附录 A：解决方案功能验证和基准测试	26
内存模式下的 Microsoft SQL Server	附录 B：Windows 系统上 Microsoft SQL 基准测试配置	31
12		
使用配置英特尔® 傲腾™ 持久内存的 Oracle 数据库构建数据仓库		
12		
经平台验证的数据分析/人工智能工作负载性能		
13		
深度学习推理		
13		
机器学习模型训练		
14		
一次构建，多处运行		
15		

简介：为什么要为您的混合云选择 VMware Cloud Foundation?

混合云基础设施兼具本地部署基础设施的优势和公有云的灵活性以及即时可用性。随着企业不断调整其现有计算资源以适应不断变化的需求和市场需要，该方法越来越受到欢迎。虽然有许多适用于公有云的服务，但有些工作负载还是更适合本地部署（例如机器学习模型需要使用敏感数据时）。因此企业越来越希望能有混合云解决方案可供选择（如图 1 所示），以实现灵活性和业务敏捷性。随着人工智能和机器学习工作负载的日益普及，混合云变得尤为重要。

利用英特尔和 VMware 提供的端到端解决方案，企业可快速启动数据库处理和人工智能，并对工作负载进行扩展以适应未来的需求。本参考架构中提出的统一的云解决方案（见下页图 2）可运行位于本地数据中心以及公有云（如 Amazon Web Services, AWS）的容器化应用和传统虚拟机。该解决方案的

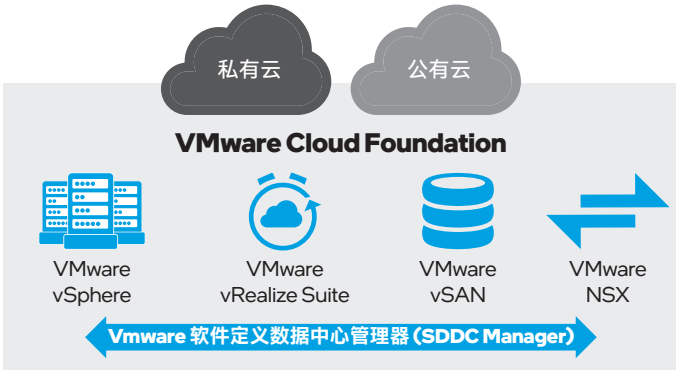


图 1. VMware Cloud Foundation 是一个在 VMware vSphere、vRealize Suite、vSAN 和 NSX 上构建的云解决方案，通过 VMware SDDC Manager 进行管理

混合云性质使企业能够扩展可用资源，并轻松地在本地和云端之间来回迁移工作负载。

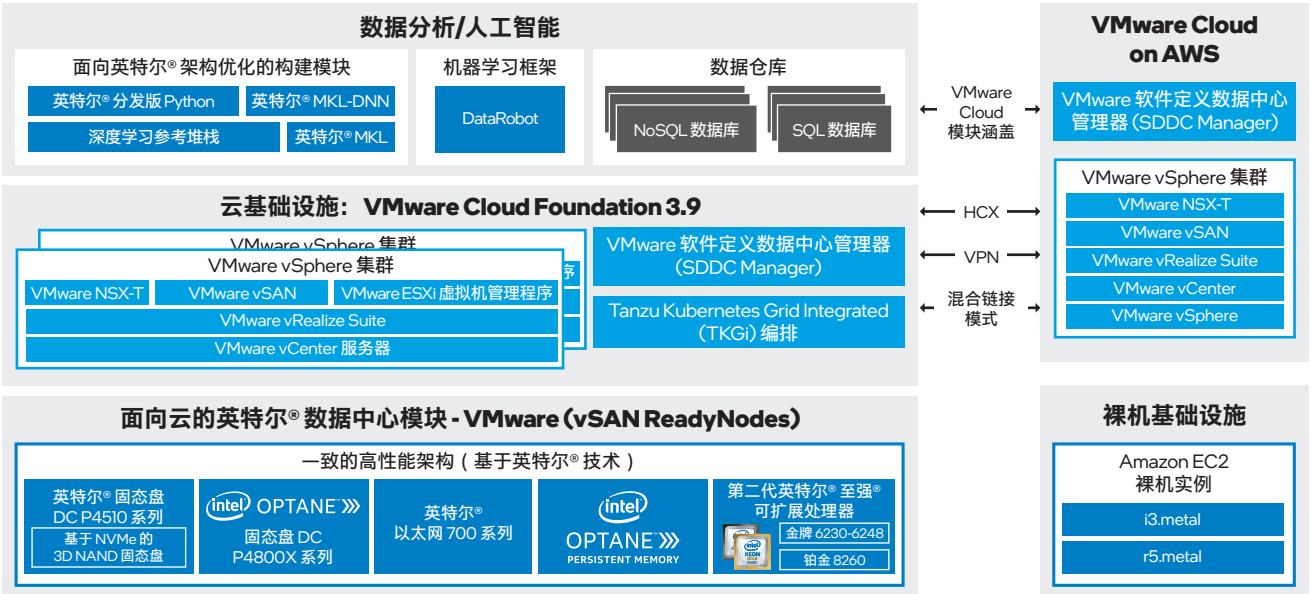


图 2. VMware 混合云平台参考架构构建模块

解决方案概述

本参考架构提供了构建混合云所需的详细配置信息。整体上，参考架构包含了优化后的英特尔® 硬件和 VMware 软件组合。

硬件概述

该解决方案的硬件堆栈建立在英特尔® 服务器主板 S2600WFOR 平台上。这些平台包括新一代英特尔® 至强® 金牌处理器和英特尔® 至强® 铂金处理器。为了实现高性能、全闪存的数据定义存储，参考架构包含英特尔® 傲腾™ 固态硬盘 DC P4800X 系列、基于 NVMe Express (NVMe) 的英特尔® 固态硬盘 DC P4510 系列和英特尔® 傲腾™ 持久内存。英特尔® 傲腾™ 持久内存引入了创新的内存技术，提供大容量的系统内存和持久性。为实现更高速的软件定义网络 (SDN)，该平台使用 25 Gb/s 英特尔® 以太网融合网络适配器 X710-DA2。

软件概述

软件方面，该解决方案由 VMware Cloud Foundation 组成，包括 VMware vSphere、VMware vSAN、VMware vRealize Suite、VMware NSX-T Data Center 和 VMware 软件定义数据中心管理器 (SDDC Manager)，以提供基础设施即服务 (IaaS) 功能。此外，基于 Kubernetes 的容器解决方案采用了 Tanzu Kubernetes Grid Integrated (TKGi)。

VMware Cloud on AWS 是该混合架构的最终目的地。VMware 混合云扩展平台 (HCX) 可在本地和云端之间实现虚拟机迁移、工作负载重新均衡和保护。除了业务持续性，它还多层应用提供了网络扩展，并且无需改变虚拟机属性。

技术介绍

本节将介绍参考架构各层中的构建模块，包括硬件、云基础设施、数据库构建模块和数据分析/人工智能构建模块。在整个堆栈中使用英特尔® 傲腾™ 技术 (如图 3 所示) 为平台带来了诸多益处。

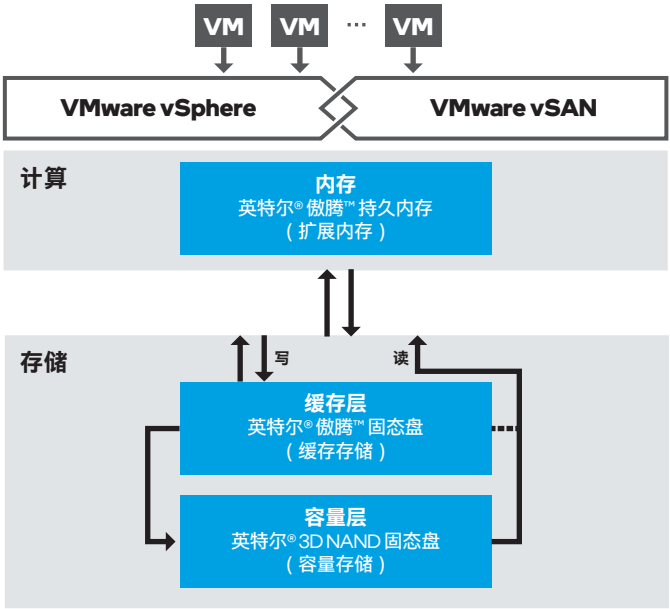


图 3. 英特尔® 傲腾™ 固态硬盘和英特尔® 傲腾™ 持久内存在架构中的位置

来源: principledtechnologies.com/VMware/VMware-HCI-Intel-Optane-VDI-0420.pdf

ⁱ 说明: TKGi 原名为 VMware Enterprise PKS。虽然本文在正文中使用了 TKGi 这个名字，但我们在整个文档中也会提及 PKS 文档，因为此次更名发生在参考架构完成之后。

英特尔® 硬件

英特尔® 傲腾™ 持久内存

英特尔® 傲腾™ 持久内存是一种全新的内存和存储技术，旨在通过提供大量具有低时延访问特性的持久内存来提高服务器的整体性能。英特尔® 傲腾™ 持久内存模组兼容 DDR4 插槽，并提供一般 DDR4 DRAM 产品所不具备的容量选择：每个模组有 128 GB、256 GB 和 512 GB 三种选择。默认的基础配置不使用持久内存，但可通过升级基础配置来使用持久内存模组，无需额外进行硬件变更，即可获得显著的性能提升¹。

如图 4 所示，英特尔® 傲腾™ 持久内存可根据业务需求和/或应用支持配置为两种不同的工作模式：内存模式和 App Direct 模式。无论采用哪种模式，持久内存都能提供传统 DRAM 模组无法提供的容量。持久内存的工作模式可通过平台 BIOS 或内存管理工具进行配置。通过内存池分区，也可实现 App Direct 双重模式。有关更多信息和指南，请访问[英特尔® 傲腾™ 持久内存快速入门指南](#)。

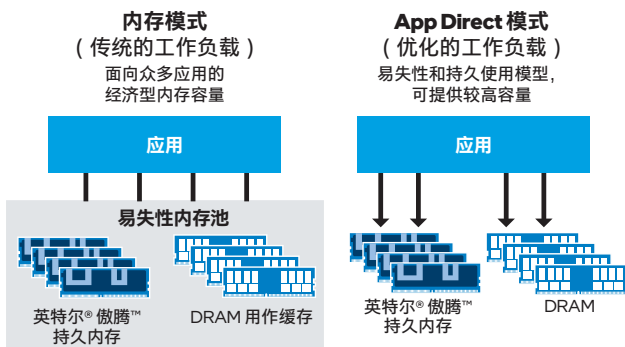


图 4. 英特尔® 傲腾™ 持久内存工作模式

来源：servethehome.com/2nd-gen-intel-xeon-scalable-launch-cascade-lake-details-and-analysis/intel-optane-dcpmm-memory-and-app-direct-modes

面向常规 DDR4 DRAM 扩展的内存模式

该工作模式非常适合扩展可用内存容量，从而在虚拟桌面基础设施 (VDI) 部署中支持更多或更大的虚拟机。这种模式还可支持更多的“热”数据，用于处理内存数据库、数据分析和其他严苛的工作负载。该模式下，企业可以将更多数据放到更靠近处理器的位置，并保持近乎 DRAM 的一致性能。持久内存作为易失性内存对所有应用（包括操作系统）可见，就像普通的 DRAM 一样。当系统关闭时，其内容会丢失。硬件方面，该模式结合了持久内存模组和标准 DRAM。DRAM 用作频繁被访问数据的缓存，而持久内存用于提供容量。通常情况下，DRAM 与持久内存的比例为 1:4 至 1:8。只有持久内存模组的大小是可见的，并可供应用和操作系统使用。由于 CPU 内存控制器负责处理所有调用并管

理用作缓存的 DRAM，操作系统或应用使用内存模式无需满足额外要求。它们并不需要识别安装了两种类型的内存。这种设置提供了更高的内存容量且成本更低。但是，在随机访问工作负载方面，这种配置可能比具有相同水平纯 DRAM 内存的系统要慢。不过纯 DRAM 内存的系统也要贵得多。

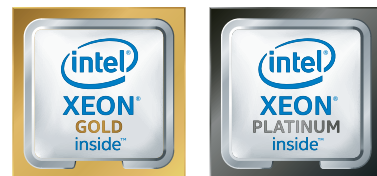
面向低时延持久内存的 App Direct 模式

App Direct 模式对内存数据库、数据分析框架和超高速存储应用等类型的工作负载更有裨益。这种模式下，DRAM 和持久内存都计入系统总内存。由于有两种类型的内存可供系统独立使用，因此该工作模式要求操作系统和应用提供额外支持。低时延操作应导向 DRAM，而需要持久性的数据或非常大的结构则应导向持久内存。在该模式下，即使系统断电，存储在持久内存模组上的数据仍具备持久性。应用可通过标准的操作系统存储调用来访问持久内存，就像访问一般存储设备上的一个普通文件系统一样（使用这种方法不需要应用识别 App Direct 模式）。该方法可明显提升速度，但持久内存还可采用和 DRAM 一样的访问方式来进一步提升性能。这种方式被称为 DAX 模式（直接访问），它不依赖文件系统调用，可显著提升速度。DAX 模式下，数据写入时间不到一微秒²。但该模式要求应用的写入方式能够使其识别持久内存。

第二代英特尔® 至强® 可扩展处理器

当前，现代企业的数据处理量在不断增加。他们需要能够满足数据分析、人工智能和内存数据库工作负载“以数据为中心”要求的计算能力。第二代英特尔® 至强® 可扩展处理器针对这些类型的应用进行了工作负载优化，每 CPU 可达 56 个内核，每路可达 12 条 DDR4 内存通道。另外，这些处理器均支持英特尔® 傲腾™ 持久内存，可实现经济的系统内存扩展。

本参考架构提供“基础”和“增强”两种配置。基础配置使用英特尔® 至强® 金牌 6248 处理器，性价比高，适合主流工作负载。增强配置使用英特尔® 至强® 铂金 8268 处理器，可有效处理高密度部署和数据密集型、时延敏感型的工作负载。需要更高性能的企业可将任一配置中的默认 CPU 替换为更高型号的 SKU。



本参考架构在基础配置中采用英特尔® 至强® 金牌处理器；在增强配置中采用英特尔® 至强® 铂金处理器

英特尔® 固态硬盘数据中心家族：英特尔® 傲腾™ 固态硬盘和英特尔® 3D NAND 固态硬盘

为了使 VMware vSAN 发挥更出色的性能，建议缓存层使用高性能英特尔® 傲腾™ 固态硬盘，容量层使用基于 NVMe 的大容量 3D NAND 固态硬盘。

英特尔® 傲腾™ 固态硬盘的独特设计提供了低时延以及每日整盘写入次数至少 30 次的耐用性³。这些特点使得它们成为大写入量缓存功能的理想选择⁴。缓存更快意味着企业能够经济高效地处理更大规模的数据集，进而挖掘重要的商业洞察。

英特尔® 固态硬盘 DC P4510 系列具备大容量，并采用了英特尔的 64 层 TLC 3D NAND 技术，与上一代的英特尔® 固态硬盘 DC P4500 系列相比，可用容量增加了一倍。这一密度提升是支持读取密集型操作的关键。此固态硬盘还提供了高可靠性和性能一致性。

面向 NVMe 驱动程序的英特尔® VMD 技术

英特尔® 卷管理设备 (英特尔® VMD) 支持通过 PCIe 总线以热插拔方式更换 NVMe 固态硬盘，且无需关闭系统，从而实现了 NVMe 固态硬盘的可维护性。此外，英特尔® VMD 还提供错误管理和 LED 管理功能 (参见图 5)。英特尔® VMD 是利用英特尔® 至强® 处理器内部的硬件逻辑实现的。VMware vSphere 6.7 借助 vSAN 或其他本地或直连存储 (DAS) 即可使用这些功能，无需安装额外的 vSphere 捆绑安装包 (VIB)。英特尔® VMD 是一个强大的 NVMe 固态硬盘热插拔解决方案，但其更大的价值在于，英特尔愿与整个生态系统分享这项技术，包括系统原始设备制造商/原始设计制造商、BIOS 程序员、PCIe 交换机供应商、固态硬盘供应商以及操作系统和独立软件供应商。

英特尔® 以太网连接和适配器

为了提升 VMware 混合云平台的性能，本参考架构采用了英特尔® 以太网 700 系列。该系列的英特尔® 以太网产品提供经过验证的性能，可满足企业在数据弹性、服务可靠性和易配置性方面的要求^{5,6,7,8}。

在物理网络层，本参考架构建议在数据平面使用两个交换机，在管理平面使用一个交换机。数据平面交换机应支持 VLAN 和巨型帧。此外，还需要一个企业级路由器解决方案来为 VMware Cloud Foundation 所要求的多个 VLAN 提供路由功能。

数据平面开发套件 (DPDK)

由英特尔开发的数据平面开发套件 (DPDK) 是一组面向英特尔® 架构优化的库和驱动程序，可用于加速数据包的处理，并且无需昂贵的定制交换机和路由器就能创建数据包转发器。它使应用开发人员通过软件和通用的英特尔® 处理器就能解决数据平面的处理需求。DPDK 能够：

- 在较少的 CPU 周期内接收和发送数据包
- 开发高速数据包捕获算法
- 运行第三方快速路径堆栈
- 提供软件预取，通过提前将数据从内存传输到缓存来提高性能

采用并受益于 DPDK 的 VMware 基础设施组件包括 VMXNET3 半虚拟化 vNIC、VDS 和直接分配功能 (Direct Path I/O 或 SR-IOV)。VMware Cloud on AWS 也支持 DPDK。

获取更多关于 DPDK 的信息，请访问[英特尔® 开发人员专区](#)。

云基础设施

VMware Cloud Foundation

VMware Cloud Foundation 通过一个私有云和公有云环境均适用的集成软件平台，提供了一条通往混合云的捷径。该平台为计算、存储、网络和安全提供全套软件定义服务，并具有以应用为核心的云管理能力，因此可在本地和“即服务”型的公有云环境中构建一个支持安全功能的简单、敏捷的云基础设施。

VMware vRealize Suite

VMware vRealize Suite 是一个面向多云的云管理解决方案，为 IT 组织提供了一个现代化平台，用于实现基础设施自动化、一致运行以及基于 DevOps 和机器学习原则的治理。

VMware SDDC Manager

软件定义数据中心管理器 (SDDC Manager) 可管理 VMware Cloud Foundation 系统的启动，创建和管理工作负载域，并执行生命周期管理，确保软件组件始终为新版本。SDDC Manager 还可监控 VMware Cloud Foundation 的逻辑资源和物理资源。

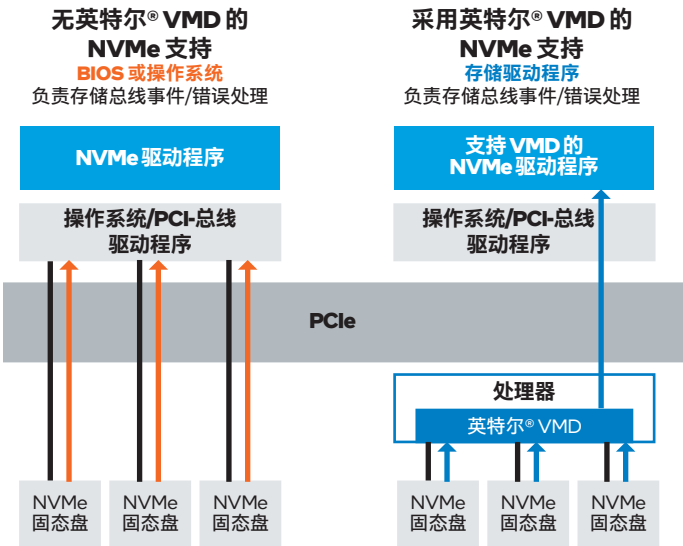


图 5. 英特尔® VMD 负责存储设备的物理管理
来源: colfax-intl.com/Servers/CX1265i-NVMe-XR7

VMware vSphere

VMware vSphere 可将虚拟化扩展至存储和网络服务，并增加了基于策略的自动化配置和管理功能。vSphere 是构建 SDDC 平台的起点。使用 Kubernetes 的 VMware vSphere 7 可简化所有企业应用的开发，提高应用敏捷性，并加快应用创新。

VMware NSX-T Data Center

NSX-T Data Center (原 NSX-T) 是网络虚拟化平台，可通过软件定义的方式启用虚拟云网络。它的工作方式类似网络虚拟机管理程序，可在软件中重现从 L2 到 L7 的全套网络服务，包括路由、交换、访问控制、防火墙、服务质量 (QoS) 以及动态主机配置协议 (DHCP)。以上所有组件可以随意组合使用，根据需求创建独立的虚拟网络。随后，这些服务可以扩展至云中和云间的各种端点。

Tanzu Kubernetes Grid Integrated (TKGi)

Tanzu Kubernetes Grid Integrated Edition (TKGi) 原名为 VMware Enterprise PKS。这是一个基于 Kubernetes 的容器解决方案，具有先进的网络、私有容器注册表和生命周期管理等功能。TKGi 简化了 Kubernetes 集群的部署和运行，使您可以在私有云和公有云上大规模运行和管理容器。在 TKGi 上，您可以使用 TKGi 控制面板来置备、运行和管理 Kubernetes 集群。

VMware Cloud on AWS

VMware Cloud on AWS 是一个混合云解决方案，可轻松实现应用的扩展、迁移和现代化，并在公有云中保护应用。该基础设施由本地使用的基于 vSphere 的同一个 SDDC 堆栈提供。VMware Cloud on AWS 包括 vSphere 和 vCenter、vSAN、vRealize、NSX-T Data Center、HCX 以及 VMware 站点恢复管理器 (SRM)，可实现混合连接和容灾即服务 (DRaaS)。一切功能都以“即服务”的形式提供给客户，方便客户快速在基于现代 Nitro 系统的 Amazon EC2 弹性裸机基础设施上部署基础设施。该解决方案利用现有的工具、流程和常见的 VMware 技术，并与 AWS 本机服务集成，因此易于采用，大幅减少了关键服务迁移到云过程中的服务中断，并且不需要再重构环境以适应公有云基础设施。

企业级基础设施以“即服务”的形式提供，SDDC 配置时间不超过两小时⁹，同时 vSAN 存储、网络、计算和安全均经过预配置。VMware Cloud on AWS 还可以根据 CPU、内存和存储需求自动调整节点规模。一般情况下，节点在几分钟就可以自动完成规模扩大或缩小。

VMware HCX

VMware HCX 是一个应用程序移动平台，旨在简化各种数据中心和云端之间的应用迁移、工作负载重新均衡并保障业务持续性。依托该平台，客户可在公有云和数据中心之间迁移工作负载，无需修改应用或配置虚拟机。该平台完全兼容 VMware 软件堆栈，使迁移更加简单、安全、可扩展。

HCX 多站点服务网络为在两个相连的 VMware HCX 站点之间迁移、扩展和保护虚拟机提供了一条安全管道 (参见图 6)。在两个站点之间的迁移过程中，它可以用于扩展 VLAN，保留 IP 和 MAC 地址，并保持现有的网络策略。在跨多个物理站点针对越来越多的复杂工作负载进行规划时，该服务网络还能够提供灵活性。

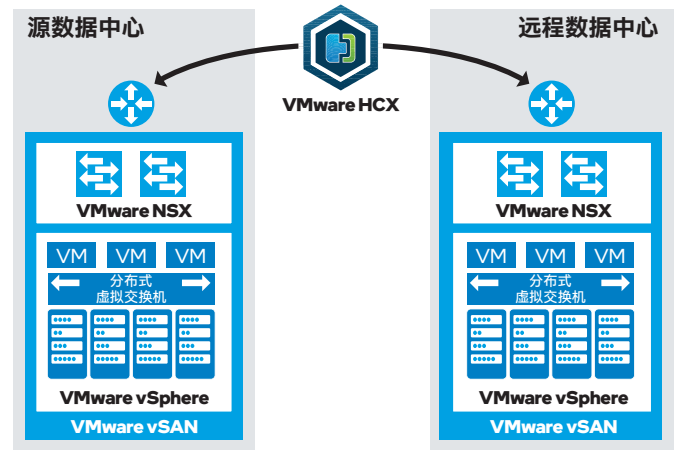


图 6. VMware HCX 概览
来源: docs.VMware.com/en/VMware-HCX/services/install-checklist/GUID-DE0AD0AE-A6A6-4769-96ED-4D200F739A68.html

数据仓库构建模块

数据仓库被认为是商业智能的核心组件之一。数据仓库是一个用来存储来自一个或多个不同来源的数据，以及当前和历史数据的中央位置。数据仓库可采用多种组织方法。该架构的主要组成部分是软硬件和数据资源，而 VMware Cloud Foundation 是部署数据仓库解决方案的出色平台 (参见图 7)。

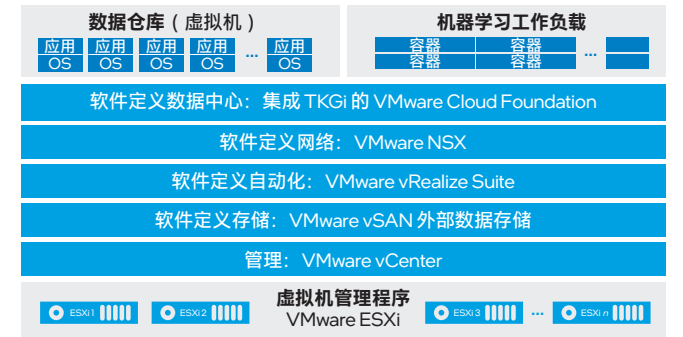


图 7. VMware Cloud Foundation 平台可用于各种数据分析、人工智能和机器学习工作负载

为了说明 VMware 混合云平台是如何支持数据仓库的，本参考架构采用了符合行业标准的主流 Oracle DB 19c 和 Microsoft SQL 2019 解决方案作为示例工作负载。除了传统的 SQL 服务，VMware Cloud Foundation 还可以容纳 NoSQL 数据库，同时还能够有效运行 Apache Hadoop 框架及其所有支持大数据和数据挖掘的相关服务。

整个平台在 vSAN 上运行，在数据冗余方面提供了额外的存储策略配置选项（可提供多种冗余级别）。平台管理员和最终用户都可以使用 vSAN（如同在 Kubernetes 部署中处理持久卷请求时那样），使整个平台存储系统尽可能得到充分利用。

数据分析和人工智能构建模块

为了保持竞争力，企业需要高性能的数据分析和人工智能。它们需要既能运行传统数据分析应用，又能运行人工智能应用的灵活解决方案。VMware 混合云平台在 VMware 基础设施中采用了多种面向英特尔® 硬件经过性能优化的组件。英特尔支持在解决方案堆栈中的多层开发机器学习工作负载。这些构建模块已经针对英特尔® 架构进行了优化，并经过了多次生产部署的验证，能够帮助企业快速实现数据分析、人工智能和机器学习工作负载的业务化。因此，企业可以即刻开始使用这些构建模块。

本参考架构展示了如何训练机器学习模型，以及如何将模型部署到混合云集群上。

英特尔® 分发版 Python

Python 是一种通用编程语言，句法简单，容易掌握。Python 还包括一个广泛的库生态系统（涵盖科学、数据转换和机器学习）。依靠这些特性，Python 既可以用于创建解决方案原型（将想法转变成可执行的代码），也能运行生产级的算法。Python 在数据科学和机器学习用例（特别是深度学习）中十分主流。英特尔提供了一款以性能为导向的分发版 Python，以加速 Python 应用。利用英特尔® 分发版 Python，企业将能够：

- 加速 Python 应用性能（与使用标准版相比），而且只需对代码稍作改动，或者根本无需改动¹⁰。
- 利用英特尔® 数学核心函数库（英特尔® MKL）、英特尔® 数据分析加速库（英特尔® DAAL）等英特尔的集成性能库，加速 NumPy、SciPy 和 scikit-learn。
- 利用英特尔® 线程构建模块（英特尔® TBB）获取新的矢量化和多线程指令，Numba 和 Cython，并实现可组合的并行性等。

详情请参见 <https://software.intel.com/content/www/cn/zh/develop/tools/oneapi/components/distribution-for-python.html#gs.6osp4w>。

英特尔® MKL

英特尔® 数学核心函数库（英特尔® MKL）面向未来的英特尔® 处理器，可大幅减少代码优化方面的工作量，并且兼容各种编译器、语言、操作系统以及链接和线程模型。该数学核心函数库随时可用，并能通过下列优势加速数学处理例程，提高应用性能，缩短开发时间：

- 具有高度优化、线程化、矢量化的数学函数，可大幅提升各款英特尔® 处理器上的性能。
- 采用工业标准的 C 编程规范和 Fortran API，无需修改代码，即可兼容各种常用的 BLAS、LAPACK 和 FFTW 函数。
- 无需创建代码分支即可自动为各处理器调度优化的代码。
- 提供优先支持服务，企业可直接联系英特尔工程师，为企业的技术问题提供值得信赖的解答。

详情请参见 <https://software.intel.com/content/www/cn/zh/develop/tools/oneapi/components/onemkl.html#gs.6otljx>。

英特尔® DAAL

英特尔® 数据分析加速库（英特尔® DAAL）是一款面向英特尔® 架构优化的构建模块库，涵盖数据分析的所有阶段，包括从数据源采集数据、预处理、转换、数据挖掘、建模、验证和决策。借助下列功能，该库可缩短自身开发高性能数据科学应用的时间：

- 高度优化的机器学习和数据分析功能。
- 同步数据获取和结果计算，以实现高吞吐量性能。
- 支持批量、流式和分布式模型使用，满足多种应用需求。
- 同一个 API 可在多个操作系统上支持应用开发。

详情请参见 <https://software.intel.com/content/www/cn/zh/develop/tools/oneapi/components/onedal.html#gs.6ot9p4>。

英特尔® 分发版 OpenVINO™ 工具包

英特尔® 分发版 OpenVINO™ 工具包可在各英特尔® 处理器上为开发人员提供出色的神经网络性能。该工具包有助于实现具备成本效益的实时视觉应用，可跨多个基于英特尔® 架构的平台实现深度学习推理，还可助力轻松实现异构执行。借助通用 API，该工具包可提供从云架构到边缘设备，以及跨各类计算机视觉加速器的实施。其可涵盖的加速器类型包括 CPU、集成 GPU、英特尔® Movidius™ 神经计算棒和英特尔® 现场可编程门阵列（英特尔® FPGA）。OpenVINO™ 工具包的功能和预优化内核有助于加快上市速度。

深度学习参考堆栈

深度学习参考堆栈（参见图 8）是一个集成的高性能开源堆栈，已针对英特尔® 至强® 可扩展处理器进行优化。该堆栈旨在帮助人工智能开发人员基于英特尔® 架构提供出色的体验。它不仅可降低深度学习软件组件中常见的复杂性，为定制化解决方案提供灵活性，还能够使企业快速创建深度学习工作负载的原型并加以部署。

新版本的深度学习参考堆栈（截至本文发布时为 DLRS V5.X）支持以下功能：

- 优化的 TensorFlow 1.X 和 TensorFlow 2.X，二者均为面向机器学习的端到端开源平台。
- 优化的开源机器学习框架 PyTorch，能够缩短从原型设计到生产部署的路径。
- 在 PyTorch 基础上进行轻量封装的 PyTorch Lightning，可帮助研究人员设定各种先进的样板训练代码。
- 转换器 (Transformer)，用于面向 TensorFlow 2.X 和 PyTorch 的先进自然语言处理 (NLP)。
- 英特尔® 分发版 OpenVINO™ 工具包，可在英特尔® 处理器上提供更高的神经网络性能，有助于实现具备成本效益的实时视觉应用。
- 采用英特尔® 高级矢量扩展 512 矢量神经网络指令的英特尔® 深度学习加速，可加速基于深度神经网络的算法。
- 深度学习编译器，一个端到端的编译器堆栈。

本参考架构演示了如何使用深度学习参考堆栈，并展示了使用面向英特尔® 架构优化的 TensorFlow 版本后所取得的性能提升。

DataRobot

本解决方案展示了 DataRobot，一个充分利用了面向英特尔® 架构优化的主流自动化机器学习平台¹¹。全球的企业和机构都在广泛地使用 DataRobot 来赋能其团队快速构建和部署机器学习模型并创建先进的人工智能应用。DataRobot 平台的算法库集合了数百个强大的开源机器学习算法，拥有众多最佳实践，在加速和扩展数据科学功能的同时也能够提高透明度、准确性并改善协作。

凭借以下功能，DataRobot 已经成为人工智能市场的热门之选：

- 针对特定问题**选择合适的模型**往往是一个繁琐且困难的过程。DataRobot 支持自动化的机器学习，能够快速高效地构建和训练数十乃至数百种算法。训练完成后，DataRobot 将按所选性能指标的顺序列出模型。为了方便模型选择，DataRobot 还会自动标记更为准确并适合部署的模型。
- 使用 DataRobot 可以轻松**校正模型**。该工具使用经过全面测试的预设设置自动运行数十个模型，以验证模型的高度准确性。如此一来，企业便可以省下这些前期工作的精力，专注于选择对他们的数据而言最准确的模型。当然，用户也可以利用 DataRobot 轻松手动校正模型。
- DataRobot 提供人性化的视觉化洞察，并自动生成模型文档蓝图，其中详细记录了建模过程中的每个步骤和所使用的算法，方便解释人工智能。企业可以使用多种工具来评估各种模型。
- 所有 DataRobot 模型都能够**立即投入生产**，且支持一键部署，快速将人工智能全面投入运作。企业可以使用集中式仪表板来监控模型，实时查看服务运行状况和使用情况；还可以管理模型准确性，轻松察觉哪些功能出现了异常，并在不中断服务的情况下部署更新。

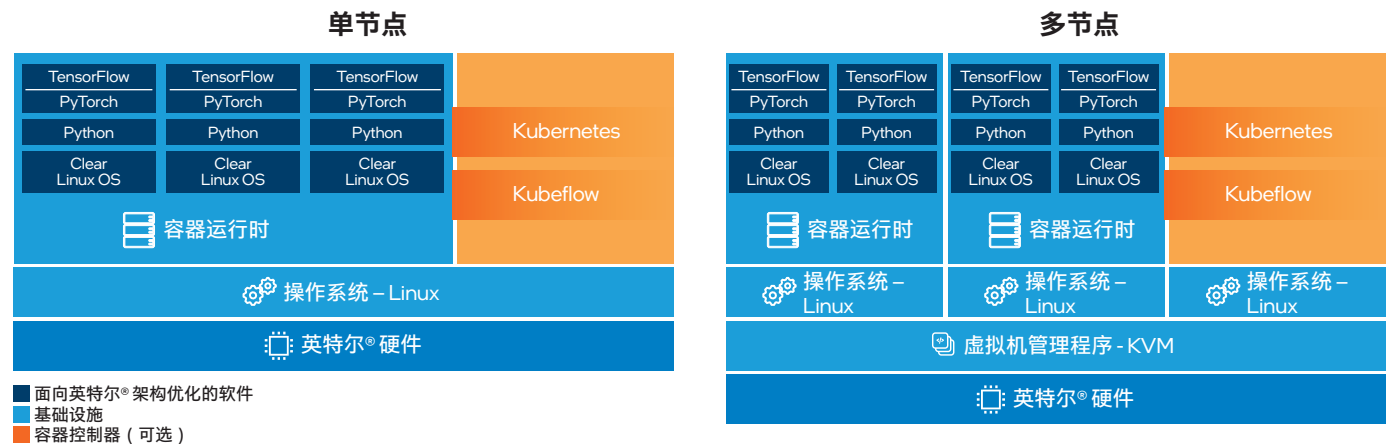


图 8. 深度学习参考堆栈加速人工智能部署

自助应用目录

本参考架构可实现快速轻松的应用部署。解决方案基于 VMware vSphere、TKGi 和 Docker 等技术。事实上，这些技术本身就代表了行业标准，已被各个社区和企业广泛采纳。因此，这一解决方案才能进一步利用开源工具和框架。解决方案包含一个叫做 Bitnami Kubeapps 的自助应用商店。这是一个 Web 应用，主要用于部署和管理 Kubernetes 集群中的应用（参见图 9）。

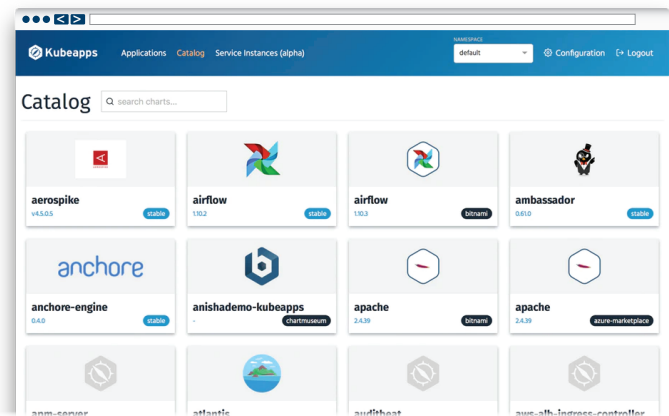


图 9. Bitnami Kubeapps 应用目录

Kubeapps 内置了一个基于 Helm 图表存储库的应用目录，可以浏览也可以部署。通过这个工具，您可以使用 Harbor 注册表添加自己的 Helm 图表存储库来管理您的应用集。这样一来，用户就可以从其 IT 部门选中并打包的应用中进行选择，或者使用各种免费提供的存储库。这种简单的方法让不懂技术的用户也能够部署和使用他们需要的应用（例如：Apache Spark 和 Jupyter Notebooks）。

关于如何使用 Kubeapps 的更多信息，请参阅附录 A：解决方案功能验证和基准测试中的“Kubeapps 的使用”一节。

经平台验证的数据仓库工作负载性能

本节将讨论使用选定数据库解决方案时，数据仓库可能存在的平台使用情况、功能和基准测试结果。

Linux 上的 Microsoft SQL Server 容器

微软官方提供了面向 Linux 上的 Microsoft SQL Server 的容器映像，让用户可以使用 Docker 引擎或 Kubernetes 部署 Microsoft SQL Server，进而快速轻松地进行应用部署。基于 Linux 的容器在 Ubuntu 16.04 基础映像之上使用 Microsoft SQL Server 2017 开发人员版本。Bitnami Kubeapps 自助应用商店中提供了用于自动化部署的 Helm 图表以及现成可用的 Microsoft SQL 应用。

有关 Linux 上的 Microsoft SQL Server 的详细信息，请参阅 Microsoft SQL Server 文档。

有了能够在容器上运行的 Microsoft SQL Server，加上自动化的 TKGi 集群和 Bitnami Kubeapp，管理员、开发人员和最终用户便有了一个强大的平台，任何人都可以借助该平台根据需求轻松部署应用和环境。

独立虚拟机上的 Microsoft SQL Server - 概述

为了提供一个数据仓库的示例，我们分别在基础配置和增强配置下对 Microsoft SQL Server 进行了测试。我们还使用了一个单独集群上的多个 HammerDB 实例来生成负载。

Microsoft SQL Server 是一个使用 Transact-SQL (T-SQL) 作为查询语言的数据库管理系统。T-SQL 是 SQL 语言的一种扩展，允许使用基本的编程结构，如变量、循环和条件指令等。出于测试目的，我们使用了 Microsoft SQL Server 2019。

HammerDB 是一个开源工具，使用特殊生成的表和虚拟用户来衡量和测试数据库的负载。该工具可以连接到不同类型的数据库引擎。虽然可以直接在 Microsoft SQL Server 上运行 HammerDB 实例，但我们建议单独创建一个 Microsoft Windows Server 实例并对 Microsoft SQL Server 数据库进行远程测试。

更多信息，请访问 HammerDB 官方网页和 GitHub。

App Direct 模式下的 Microsoft SQL Server - 配置和基准测试结果

App Direct 模式下的持久内存可在电源循环中保留存储在其中的内容，即使意外断电或系统崩溃导致系统电力中断时也是如此。App Direct 模式有两种使用方法：

- **块访问模式：**在此配置中，持久内存的工作方式类似于标准存储。所有数据请求都要经过常规存储堆栈（操作系统仅将持久内存视为一块硬盘）。因此，存储空间的速度非常快，可供任意应用使用。
- **直接访问 (DAX) 模式：**在此配置中，持久内存模组的工作方式与 DRAM 类似，它能够绕过操作系统的存储堆栈，从而尽可能降低时延。但是，应用必须支持 DAX 才能获得这一额外的性能优势。Microsoft SQL Server 2019 就是这样一款应用。DAX 模式是我们测试的重点。

这两种模式下，持久内存都可以用作数据仓库的存储，用于实现系统的低时延和高速读写。如果需要更高性能，应当尽可能使用 DAX 模式，而不是块访问模式。另一方面，如果需要较高的存储性能，但是应用却不支持 DAX 模式，则块访问模式就会因其应用兼容性而成为理想选择。尽管已经有越来越多的应用直接支持 DAX 模式的持久内存，但与传统解决方案相比，块访问模式还是具有灵活、易用和高性能的优势。

本参考架构测试示例使用的工作负载基于两种主要的测试场景：

- **基础场景**在普通的入门级英特尔® 架构平台上执行，为各类工作负载提供了一个良好起点。但是，该平台未安装持久内存。我们将此平台称为基础集群。
- **增强场景**基于先进的英特尔® 至强® 铂金处理器，配备大容量驱动器以及使用 App Direct 模式的持久内存模组。该平台展现了本参考架构的较高潜能。我们将此平台称为增强集群。

需要注意以下几点：

- 增强集群节点上的可用持久内存存储空间高于 Microsoft SQL Server 实例的实际所需（对于四节点集群而言，利用率应低于 75%）。这种过度预配的方法会使维护工作和虚拟机迁移变得非常容易。以 App Direct 模式存储的数据实际位于托管该特定 Microsoft SQL Server 虚拟机的 VMware ESXi 节点上。如果要迁移该虚拟机（例如，出于负载均衡或维护目的），目标 ESXi 节点的持久内存中必须有足够的可用空间来存储所迁移的虚拟机上所有 App Direct 模式下存储的数据。
- 持久内存模组上的数据分布不同于 vSAN。如果节点因某种原因宕机，在节点重启前数据将无法访问。

- VMware 没有为 App Direct 模式下存储的数据采取任何复制机制。如果节点硬件出现故障，数据将会丢失。因此，企业的生产部署必须使用 Microsoft SQL Server 支持的复制或备份方法来实现高可用性。

如上所述，DAX 模式是一种高速机制，并受到 Microsoft SQL Server 的支持。然而，并不是所有文件类型都支持 DAX 模式。数据库应拆分为多个文件。数据库文件存放在配置为 DAX 模式的持久内存上，日志文件存放在配置为块访问模式的持久内存上。

Windows 和 Linux 版本的 Microsoft SQL Server 2019 中都引入了混合缓冲池功能（参见图 10）。该功能支持引用驻留在持久内存上的 SQL 数据库文件中的数据页，而不必将数据页复制到基于 DRAM 的缓冲池中。持久内存利用内存映射的 I/O，无需使用操作系统的 I/O 堆栈即可访问数据页，因此提供了性能优势。有关如何在 Microsoft SQL Server 实例上启用混合缓冲池功能，请参阅 [Microsoft 提供的操作指导](#)。

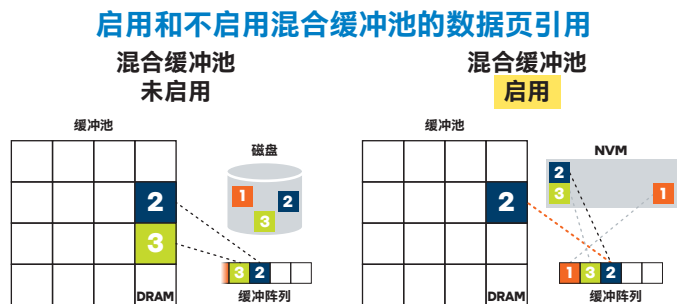


图 10. 混合缓冲池功能通过改变内存数据的寻址方式提升了 I/O 性能
来源: docs.microsoft.com/zh-cn/sql/database-engine/configure-windows/hybrid-buffer-pool?view=sql-server-ver15

在开始 Microsoft SQL Server 基准测试前，您必须进行适当的存储配置。为了获得理想的结果，我们建议将 Microsoft SQL Server 数据库分布到八个磁盘，具体如下所示：

- 四个磁盘ⁱⁱ用于存储数据文件，每个磁盘 18 GB，将数据拆分为 16 个文件，平均分布在四个磁盘上（每个磁盘四个数据文件）
- 两个磁盘ⁱⁱ用于存储 TempDB 文件，每个磁盘 200 MB
- 一个磁盘ⁱⁱ用于存储 TempLog 文件，大小为 100 MB
- 一个磁盘ⁱⁱ用于存储事务日志，大小为 10 GB

上述建议的磁盘大小适合 750 个仓库的测试场景。您可遵循本文后续“[为英特尔® 傲腾™ 持久内存准备虚拟机](#)”一节中的建议来为虚拟机部署持久内存模组 (NVDIMM)。创建完成后，每个数据磁盘会格式化为 NTFS 并配置为支持 DAX 模式，而日志磁盘则格式化为 NTFS 并配置为支持块访问模式。有关 NVDIMM 驱动器使用的

ⁱⁱ 注：在本例中，“磁盘”指的是持久内存上的独立卷。

详细列表，请参阅附录 B: Windows 系统上 Microsoft SQL 基准测试配置。

测试方法

我们按照上述配置在虚拟机上进行了基准测试。对于每个 Microsoft SQL Server 虚拟机，都有一个安装了 HammerDB 的单独虚拟机，用于生成负载。我们将 HammerDB 安排到了另一个集群上，避免对 Microsoft SQL Server 的资源使用造成干扰。我们通过增加集群中此类虚拟机的数量来进行扩展，从而增加集群中存储和负荷的数据仓库总数。

在两个集群中，虚拟机都配备 8 个虚拟 CPU (vCPU) 和 44 GB 的 DRAM。对于基础集群，我们先在单个 ESXi 节点上部署五个虚拟机，然后在集群的第二个节点上再部署五个虚拟机，依次迭代，直到所有四个节点上共部署 20 个虚拟机。Microsoft SQL Server 虚拟机绑定到特定主机，其他非 Microsoft SQL Server 的虚拟机均启用分布式资源调度程序 (DRS)。在选择 vCPU 数量、DRAM 大小和虚拟机数量时，原则是尽可能地提高解决方案的可扩展性和稳定性。集群内的其他服务 (包括 vSAN、NSX-T Edge 虚拟机和 TGKi) 共享 DRAM 和 CPU 资源。过度分配 DRAM 或 CPU 资源会对 vSAN 性能产生负面影响，增加读写时延，进而影响所有其他服务的性能。

增强配置拥有额外的 CPU 资源和持久内存，因此可以在每个 ESXi 节点上部署更多 Microsoft SQL Server 虚拟机实例——每个节点部署了八个实例。因此，大部分数据以 App Direct 模式存储在 ESXi 主机本地，减轻了 vSAN 的负载。为了确保整体内存使

用率低于 90%，我们必须减少每个虚拟机的 DRAM 大小。这保证了操作系统、vSAN 和 NSX-T 服务不间断运行。

基础和增强集群的基准测试结果

即使容量增加，增强集群的性能仍优于基础集群：与基础集群所取得的最优结果相比，增强集群的每分钟事务处理量 (TPM) 高达前者的 3.34 倍 (参见图 11) ¹²。

我们还探索了这两种集群的时延以及服务级别协议 (SLA) 相关值的达标情况。这些值包括“CPU 时间：请求”和“CPU 时间：总时间”。两个集群上的所有测试场景中，至少 98% 的请求在 CPU 内的时间小于 5 毫秒 (图 12 中的橙色线) ¹³。“CPU 时间：总时间”的测试结果更能说明问题。基础集群无法满足“80% 的请求在 CPU 内的总时间小于 20 毫秒”的 SLA 值要求。反观增强集群，几乎所有批次都达到此 SLA 值的要求，仅在资源消耗最高的场景中略有下降¹⁴。

App Direct 模式总结

基准测试结果表明，在增强集群上，我们不仅能够在每个 ESXi 节点上运行更多 Microsoft SQL Server 实例和数据仓库，密度提升高达 1.6 倍，而且整体平台性能也更出色，TPM 提升高达 3.34 倍。同时，基础集群通过充分利用 VMware Cloud 基础设施也可以提供灵活性、高可用性和容灾能力。本参考架构最初发布时，这些功能还不能与持久内存模组一起使用。鉴于此限制，配置了 App Direct 模式的虚拟机也无法实现实时迁移。因此，增强集群在虚拟环境中使用持久内存的 App Direct 模式所获得的高性能只有在本地才能实现。

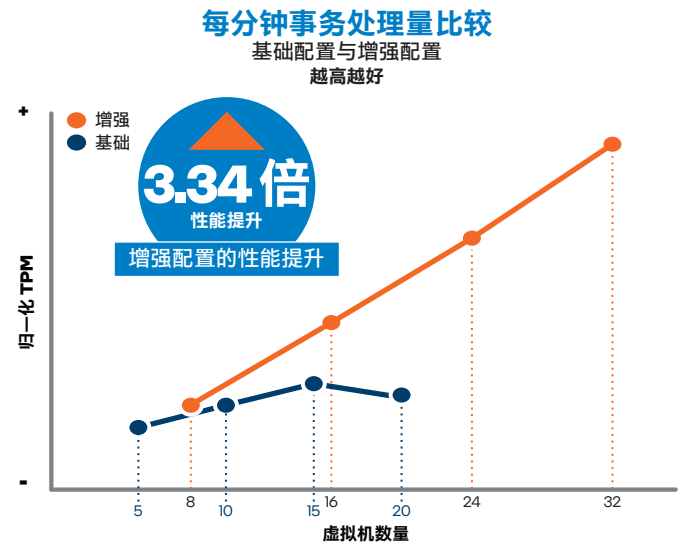


图 11. 增强集群的 TPM 高达基础集群的 3.34 倍，且具有高度可扩展性

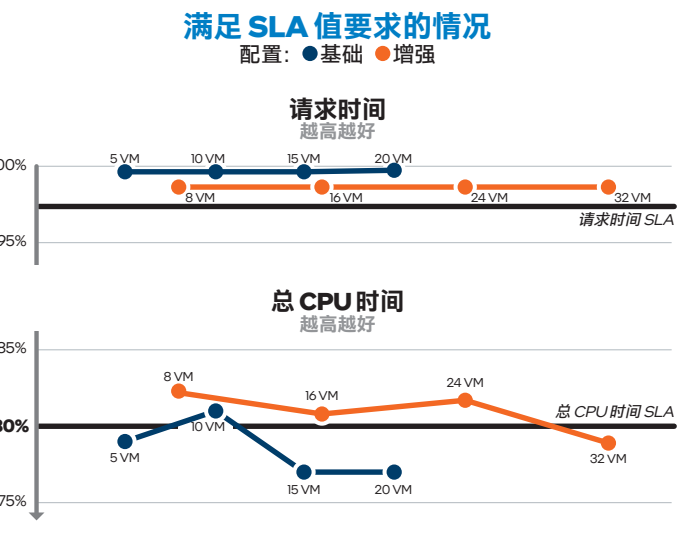


图 12. 在增加虚拟机的过程中，增强集群能够持续满足 SLA 值的要求，而基础集群无法满足要求

内存模式下的 Microsoft SQL Server

物联网 (IoT)、机器学习和人工智能技术通常需要快速访问大型数据集。在扩展内存数据库部署时，常规 DRAM 的大小限制可能成为瓶颈。此时，可以通过与集群共享存储来作为替代方案，但是这样会增加总成本，并导致一定程度的性能损失，使管理变得更加复杂。英特尔® 傲腾™ 持久内存存在内存模式下可通过整合来提高硬件的使用效率，从而提供高内存密度。在该模式下可以对现有环境进行纵向扩展而不是横向扩展。当单服务器节点有了更多可用内存，企业和机构就可以将较小的纯 DRAM Microsoft SQL Server 虚拟机整合到配备 DRAM 和持久内存的更少节点中。随着数据库的增长，他们可以使用更大的内存升级这些服务器。每路持久内存的大小可高达 4.5 TB，配备 512 GB 持久内存模组的双路系统可以在一个双路服务器上配置 6 TB 内存。内存密度提升能够实现以接近 DRAM 的速度存储和访问更多数据，最终以低时延访问更大量的数据。

由于启用 DAX 的 App Direct 模式能够显著提升 Microsoft SQL Server 的性能，因此本参考架构没有对 Microsoft SQL Server 进行内存模式下的基准测试。相比之下，内存模式能为大规模工作负载提供更大容量和密度，实现近似 DRAM 的性能¹⁵。但需要记住的是，使用内存模式时，您还可以构建仅使用 DRAM 的设置无法复制的硬件配置。

使用配置英特尔® 傲腾™ 持久内存的 Oracle 数据库构建数据仓库

Oracle 为企业提供了业内出众的数据库平台。许多大型企业和机构使用 Oracle 数据库作为信息管理平台的基础。许多客户已经使用 VMware ESXi 和 vSphere 对资源密集型工作负载进行充分虚拟化，从而成功部署了基于 Oracle 的数据仓库。本参考架构描述了 VMware 上的 Oracle 设置示例，以此说明在 vSphere 上运行 Oracle 的优势以及它在管理大型数据集方面的高效。

简便的可管理性

虚拟化的 Oracle 环境通过以下方式提高可管理性：

- **整合：**VMware 技术可以在几乎不影响整体性能的情况下，将多台应用服务器整合到一台物理服务器上。
- **易于置备：**VMware 虚拟化技术可将应用封装成可复制或移动的映像，大大降低了应用置备和部署的成本。此外，还可以交付多种 Oracle 数据库实例（包括不同版本的 Oracle 数据库；面向不同用途的实例，如开发、质检或生产；或面向不同最终用户的实例等）。
- **工作负载迁移：**使用 vSphere vMotion 功能，客户能够以较短的停机时间将工作负载从源 ESXi 主机实时迁移到目标 ESXi 主机，这也可以简化常见的硬件维护等工作。
- **纵向扩展：**借助 vSphere，虚拟机可以轻松利用多核处理器。必要时，可以增加 vCPU 来提升虚拟机的计算能力。
- **高可用性：**借助分布式资源调度程序，集群内的虚拟机开箱即可实现自动故障转移和负载均衡。负载均衡有助于提高集群的资源利用率，甚至能够大幅降低节点完全故障所带来的负面影响，因为虚拟机可以自动从故障节点转移到健康系统并启动。

在内存中处理大型数据集

随着越来越多的数据存储到现代数据仓库中，企业和机构需要能进行相应扩展的解决方案。一种常见的扩展方法是构建集群并将数据分布到系统各处，然而，这种方法无论是在性能上还是总成本上都不尽如人意。

Oracle 数据库可以配置为使用内存模式或 App Direct 模式的英特尔® 傲腾™ 持久内存。内存模式下，Oracle 数据库可以访问 1.5 TB、3 TB 甚至 6 TB 的内存来执行内存操作，例如内存连接、内存聚合和内存列格式数据存储等。App Direct 模式下，Oracle 数据库可以将持久内存用作快速磁盘驱动来存储 +DATA 和 +REDO 文件。

经平台验证的数据分析/人工智能工作负载性能

以下部分将讨论深度学习推理和机器学习模型训练工作负载。

深度学习推理

图像分类是深度学习的一个常见用例。我们使用深度学习参考堆栈中的 TensorFlow 分发版和英特尔® Model Zoo 的预训练模型对 ResNet50 v1.5 和 Inception v3 拓扑进行了基准测试。有关该基准测试的详细说明，请参阅附录 A：解决方案功能验证和基准测试中的“使用深度学习参考堆栈容器运行 TensorFlow 基准测试（再现推理基准测试）”一节。

我们进行了两次实验，第一次使用胖虚拟机，第二次使用包含多个工作虚拟机的 TKGi 实例作为较小的虚拟机。两次实验都使用通过 VMware 软件提供的整个物理节点。基础配置的胖虚拟机配备 80 个 vCPU，而增强配置则配备 96 个 vCPU。基础配置和增强配置的 Kubernetes 集群均使用 6 个工作虚拟机，每工作虚拟机配备 16 个 vCPU。

胖虚拟机的测试结果

在胖虚拟机实验中，我们对比了深度学习参考堆栈容器的吞吐量和标准 TensorFlow 容器的吞吐量。在基础配置中，以 fp32 精度对 ResNet50 v1.5 拓扑进行基准测试，结果发现使用深度学习参考堆栈容器时的性能相较于标准 TensorFlow 容器提升 2.2 倍；在增强配置中，性能提升 2.5 倍（参见图 13）¹⁶。同样以 fp32 精度对 Inception v3 拓扑进行基准测试，结果发现在基础配置中，使用深度学习参考堆栈时的性能相较于标准 TensorFlow 提升 2.4 倍；在增强配置中，性能提升 3 倍（参见图 14）¹⁷。这些结果表明使用面向英特尔® 架构优化的软件具有显著效果。

这些数字说明，使用深度学习参考堆栈容器时，ResNet50 v1.5 和 Inception v3 拓扑的吞吐量均有大幅提升。该测试证实了深度学习参考堆栈容器的优化在提高英特尔® 处理器利用率方面的有效性。

Kubernetes 测试结果

在由 TKGi 配置的 Kubernetes 集群中，我们对 1 到 6 个并行的工作虚拟机的深度学习工作负载进行基准测试，以此测量深度学习参考堆栈容器的吞吐量扩展。结果发现当有更多作业并行运行时，吞吐量提升效果十分显著，证明了 ESXi 调度程序的有效性。在多节点系统中，运行工作负载的虚拟机扩展后，Inception v3 拓扑的整体吞吐量也会随之高效扩展¹⁸。图 15 显示了增强配置中扩展效率的显著提升。

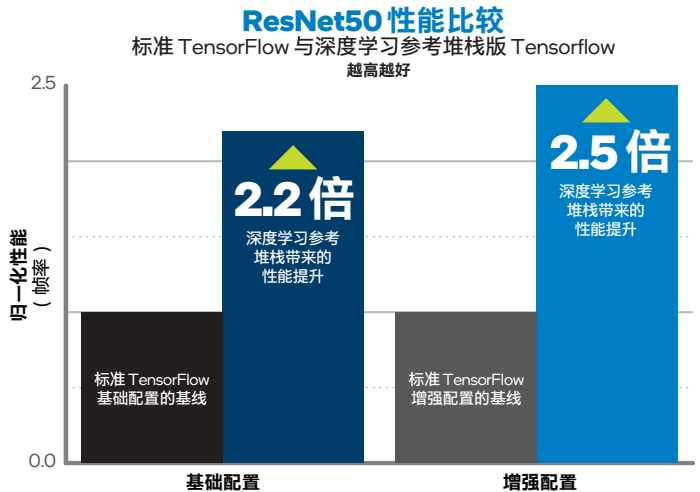


图 13. 使用深度学习参考堆栈版本的 TensorFlow 后，ResNet50 v1.5 拓扑在基础配置和增强配置上的吞吐量均达到了原来的两倍以上

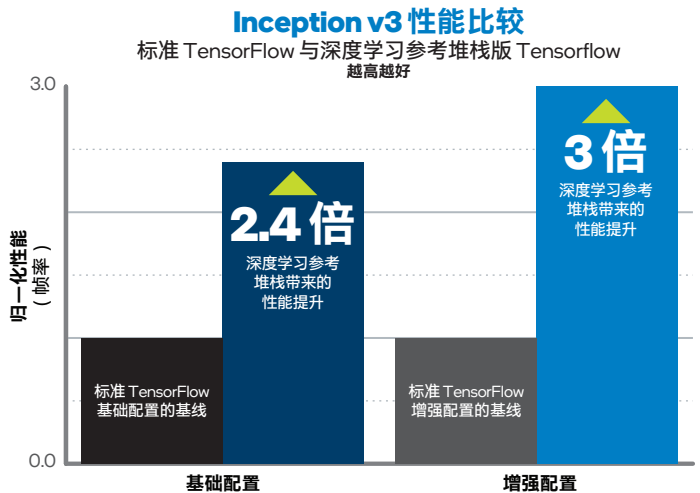


图 14. 使用深度学习参考堆栈版本的 TensorFlow 后，Inception v3 拓扑在基础配置上的吞吐量达到了原来的两倍以上，而在增强配置中达到三倍

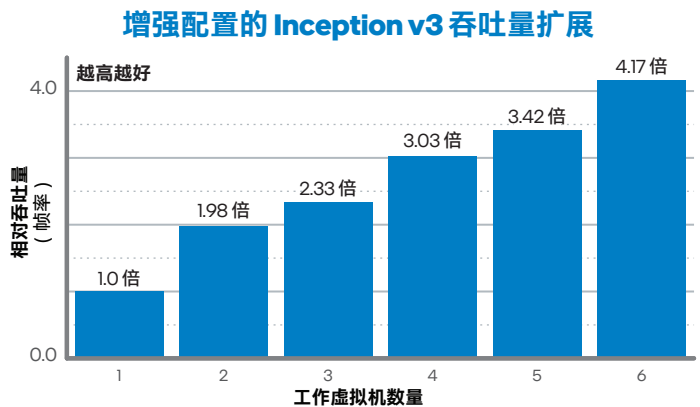


图 15. 增强配置中使用深度学习参考堆栈容器并行运行 1 至 6 个工作虚拟机，以 fp32 精度对 Inception v3 拓扑进行测试得到的吞吐量扩展

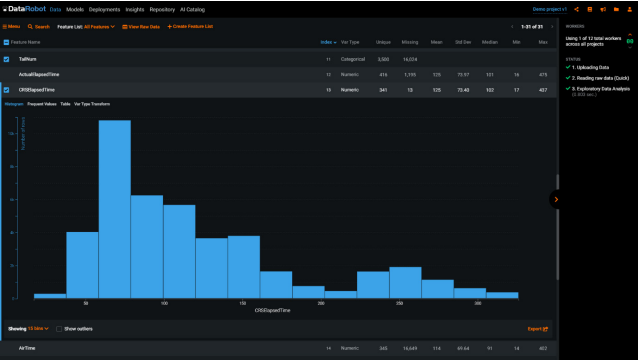
机器学习模型训练

本节工作负载示例的主要目标是展示如何快速便捷地训练多种简单模型，从而对重要结果做出预测。DataRobot 这样的先进工具能较过去大大简化这一过程。

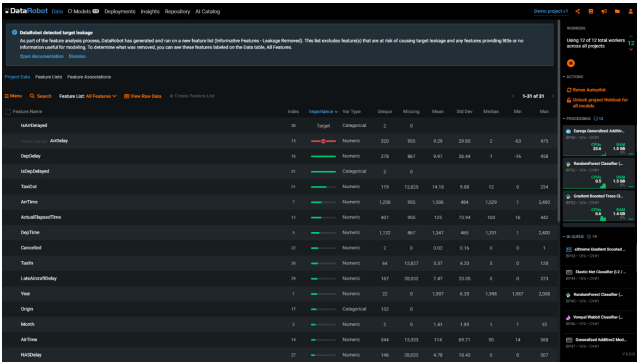
- 1. **上传数据：**DataRobot 允许选择 JDBC 源，或者从 URL、Apache Hadoop 分布式文件系统 (HDFS) 或本地存储的文件上传数据。该工具可以处理 .csv、.tsv、.dsv、.xls、.xlsx、.sas7bdat、.parquet、.avro、.bz2、.gz、.zip、.tar 和 .tgz 格式的文件。广泛的选择可提供更好的用户体验。

本演示所选的数据集为航班数据，具体可参见[此处](#)。

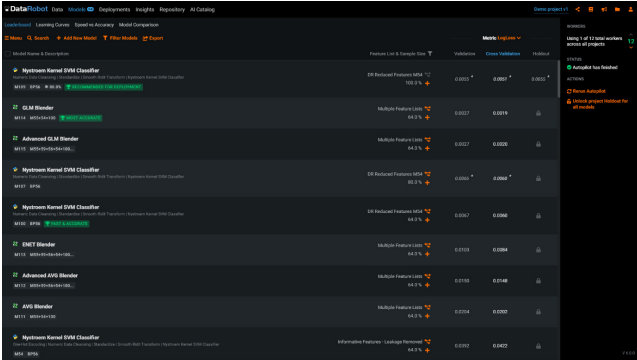
- 2. **浏览人工智能目录：**上传的数据集将显示在“人工智能目录”中。这里，您可以看到基本数据分析和数据集摘要，以及基本信息和特征列表。
- 3. **浏览并可视化数据：**方便的数据可视化是为了让用户更好地理解数据，便于用户选择重要特征数据用于训练。DataRobot 会自动浏览数据集并识别变量类型和统计数值，例如均值、中位数和标准差等。单击特征名称即可查看详细信息。



- 4. **创建新项目：**创建一个新项目，开始数据处理。DataRobot 会自动分析数据并添加建议。例如，DataRobot 可以根据现有特征创建新字段。
- 5. **开始训练流程：**输入名称，选择目标特征。系统会根据字段类型（例如，分类或回归）自动识别问题。单击“Start”按钮；Autopilot 会让 DataRobot 选择要检查的训练算法（蓝图）。
- 6. **优化模型：**DataRobot 的算法会再次分析数据并检查数据冗余，或者排除可能导致目标泄漏风险的一个或多个特征，以及排除对建模作用极小或无效的特征。该平台还会决定哪些特征对结果最重要。这些过程的进度显示在右栏中。



- 7. **比较算法：**DataRobot 使用不同的算法训练模型，并对其加以比较以确定哪种算法得到了更理想的结果。但是，该平台不会使用整个数据集对所有模型进行训练，而是根据部分数据集的结果和性能来确定哪些算法更有效，然后利用这些算法继续训练。这种方式可以节省时间和资源。用户可以对算法进行比较。数据在训练期间也会实时更新。训练完成后，DataRobot 会为通过标记来显示其推荐，如“推荐用于开发”、“结果准确”和“快速且准确”等，以此帮助用户选择算法。



- 8. **部署模型：**所有模型均可下载和手动部署。用户还可选择自动部署模型，这样可以快速启动预测。部署完成后，用户界面上可以找到使用该模型所需的所有数据。
- 9. **开始预测：**DataRobot 提供可用于推理的 REST API。用户可以利用该 REST API，通过多种编程语言使用模型。此外，用户还可以利用自动生成的 Python 脚本，而不必直接调用 API。这些代码可在用户界面上找到，并按需进行调整。代码可以复制并用来构建更大的系统，也可以原封不动地使用。
- 10. **监控模型健康：**用户可以在用户界面上观察模型行为，追踪模型做出的预测数量、请求、数据错误率和模式，也可以追踪数据漂移。

一次构建，多处运行

本文中描述的机器学习用例包含以下元素：

- 使用 DataRobot 进行模型训练
- 在内部应用目录中发布模型（推理引擎，评分器）
- 部署评分器 (scorer)

上一节“机器学习模型训练”中已经介绍了模型训练的过程。完成此步骤后，就可以得到 Python 或 Java 运行时的模型二进制文件（评分器）。

包含这些二进制文件的 Docker 映像会自动构建并发布到容器注册表——本示例使用的是 Harbor 注册表。作为云原生计算基金会孵化的项目，Harbor 具有较好的合规性、性能和互操作性，能够帮助企业跨云原生计算平台（如 Kubernetes 和 Docker）统一安全地管理映像。由于评分器（或任何其他应用）可以在该注册表中访问，因此用户可以自选部署环境，本地集群、私有云或公有云均可。

将评分器发布到 Kubeapps 等应用目录有助于改善企业用户和平台操作员的用户体验，因为这样一来，他们只需点击几下鼠标即可在 Web 用户界面上轻松快速地部署评分器。此外，目录可以与容器注册表集成，并使用 Webhook 自动检测注册表中是否出现了新版本应用。这种情况下，目录会用新版本应用替换当前版本，无缝地重新部署应用。

DataRobot v6.0.0 中的预测服务器功能简化了预测模型部署。通过此功能，用户可以在用户界面部署预测服务（训练完成后，用户可以选择部署所选模型）。然后，用户就可以通过 HTTP 调用或使用 Python¹⁹ 或 R²⁰ 客户端来获取预测服务。

物料清单

硬件

本参考架构可以从 8 个服务器的单机架扩展到 15 个域（一个管理域和 14 个工作负载域），最多可包含 400 个 ESXi 服务器。参考架构在单机架上使用 12 台英特尔® 服务器、两台架顶式 Arista 交换机和一台云达科技管理交换机，需要时也可以添加额外的机架。

生成初始软件映像需要一台额外的运行虚拟化软件的服务器或笔记本电脑和一台私有交换机。这些组件不是参考架构的组成部分，在完成 VMware Cloud Foundation 映像生成和启动过程后也不再需要这些组件。

为了演示对异构硬件配置的支持，本参考架构使用了两种类型的服务器，它们在 CPU 型号、内存大小和驱动器数量方面有所不同。企业可以在一定程度上修改基础 vSAN ReadyNode 配置，添加更多内存或驱动器，或将 CPU 替换成拥有更多内核或更高时钟速度的型号。您可单击[此处](#)查看一般规则。

在本参考架构中，每台机架由两个架顶式交换机和一个带外管理交换机组成。在多机架部署中，建议再增加一组脊交换机（通常安装在第二个机架）。随着 VMware Cloud Foundation 3.0 和 Bring Your Own Network (BYON) 项目的面世，VMware 不再认证交换机与 VMware Cloud Foundation 的兼容性。对于网络组件，本参考架构使用了两种型号的网络交换机：数据平面使用两台 Arista DCS-7060CX2-32S-R 交换机，管理平面使用一台 Arista DCS-7010T-48-R 交换机。您也可以改用任何其他性能相似的网络硬件。

表 1 列出了本参考架构的硬件组件。

表 1. 参考架构硬件组件

管理集群 4 个节点			基础工作负载域 4 个节点			增强工作负载域 4 个节点		
	部件描述	数量 (每节点)		数量 (每节点)	部件描述		数量 (每节点)	
基础 SKU	英特尔® 服务器系统 VRN2208WFAF82R	1	英特尔® 服务器系统 VRN2208WFAF82R	1	英特尔® 服务器系统 VRN2208WFAF83R		1	
主板	英特尔® 服务器主板 S2600WFOR	1	英特尔® 服务器主板 S2600WFOR	1	英特尔® 服务器主板 S2600WFOR		1	
CPU	英特尔® 至强® 金牌 6230 处理器 (20 核, 2.1GHz)	2	英特尔® 至强® 金牌 6248 处理器 (20 核, 2.5 GHz)	2	英特尔® 至强® 铂金 8268 处理器 (24 核, 2.9 GHz)		2	
内存	32 GB RDIMM DDR4-2933	12	32 GB RDIMM DDR4-2933	12	32 GB RDIMM DDR4-2666		12	
					128 GB 英特尔® 傲腾™ 持久内存模组		12	
缓存层	375 GB 英特尔® 傲腾™ 固态硬盘 DC P4800X 系列 (PCIe x4 U.2)	2	375 GB 英特尔® 傲腾™ 固态硬盘 DC P4800X 系列 (PCIe x4 U.2)	2	375 GB 英特尔® 傲腾™ 固态硬盘 DC P4800X 系列 (PCIe x4 U.2)		2	
容量层	2 TB 英特尔® 固态硬盘 DC P4510 系列 (2.5" NVMe U.2)	4	2 TB 英特尔® 固态硬盘 DC P4510 系列 (2.5" NVMe U.2)	4	2 TB 英特尔® 固态硬盘 DC P4510 系列 (2.5" NVMe U.2)		6	
引导设备	480 GB 英特尔® 固态硬盘 DC S4510 系列 (M.2, 80 mm)	1	480 GB 英特尔® 固态硬盘 DC S4510 系列 (M.2, 80 mm)	1	480 GB 英特尔® 固态硬盘 DC S4510 系列 (M.2, 80 mm)		1	
网卡	英特尔® 以太网融合网络适配器 X710-DA2	1	英特尔® 以太网融合网络适配器 X710-DA2	1	英特尔® 以太网融合网络适配器 X710-DA2		1	

软件

本参考架构主要包含两套软件组件：VMware Cloud Foundation 和 TKGi。后者需要多个组件和支持基础设施。此外，要实现与外部网络的无缝集成和全球时间同步，还需要企业网络时间协议 (NTP) 服务器和域名系统 (DNS) 服务器等多种网络服务。表 2 和表 3 提供了软件组件信息。有关相关要求和前提条件的完整列表，请参阅 [VMware 官方文档](#)。

表 2. VMware Cloud Foundation 产品和服务

组件	版本	内部版本
VMware Cloud Foundation 捆绑安装包	3.9	14866160
Cloud Builder 虚拟机	2.2.0.0	14866160
VMware ESXi 虚拟机管理程序	ESXi 6.7 Update 3	15160138
VMware vSAN	6.7 Update 3	14263135
VMware NSX Data Center for vSphere	6.4.5	13282012
VMware NSX-T Data Center	2.5	14663974
VMware vCenter Server 设备	vCenter Server 6.7 Update 3	14367737
VMware SDDC Manager	3.9	14866160
VMware vRealize Suite Lifecycle Manager	2.1 Patch 2	14062628
Tanzu Kubernetes Grid Integrated (TKGi)	1.5	14878150

表 3. 其他软件组件

组件	版本
Kubeapps	1.10.0
TensorFlow 映像	tensorflow/tensorflow:1.15.0-py3
深度学习参考堆栈分发版 TensorFlow 映像	clearlinux/stacks-dlrs-mkl:v0.5.0
Oracle 数据库	19.3
DataRobot	6.0.0
Microsoft SQL Server 2019	15.0.2070.41
Microsoft Windows Server 2019 Datacenter	17763.rs4.180914-1434
Windows HammerDB	3.3

注：

- ESXi 虚拟机管理程序的内部版本比官方文档中的版本高。在部署 VMware Cloud Foundation 之前，ESXi 系统已安装了额外的安全补丁。
- 表 2 仅列出了主要组件。有关其他组件的信息，请参阅 [VMware Cloud Foundation 发行说明](#)。

BIOS 和固件组件

表 4 从硬件角度列出了本解决方案使用的固件和驱动程序版本。

表 4. BIOS 和固件组件

固件/驱动程序名称	版本
BIOS	02.01.0010
BMC	2.37
ME	04.01.04.339
SDR	1.98
NIC 固件	6.80 0x8003d051.2007.0
NIC 版本	1.9.5
英特尔® 傲腾™ 固态硬盘 DC P4800X	E2010435
英特尔® 固态硬盘 DC P4510	VDV10170
英特尔® 傲腾™ 持久内存固件	01.02.00.5417
CPU 微代码	基础： 0x0500002c
	增强： 0x0500002c
	管理： 0x0400002c

部署模块

部署 VMware Cloud Foundation、NSX-T、TKGi 和 vSAN 等解决方案是为了将传统数据中心改造成软件定义数据中心 (SDDC)，让管理员能够根据最终用户的实际需求来定义、部署和管理集群和资源。其中涉及的每个组件都是独立的产品，可以单独使用。

VMware Cloud Foundation

VMware Cloud Foundation 是一个集成软件平台，可在标准化的超融合架构上自动进行完整的 SDDC 部署和生命周期管理。VMware Cloud Foundation 由以下核心组件组成 (参见图 16) :

- 用于计算虚拟化的 VMware
- 用于网络虚拟化的 VMware NSX for vSphere 和 NSX-T Data Center
- 用于存储虚拟化的 VMware vSAN
- 用于云监控的 VMware vRealize Suite

VMware Cloud Foundation 使企业和机构能够构建企业直接可用的面向私有云和公有云的云基础设施。

VMware Cloud Foundation 的标准架构模型包括一个用于所有管理组件的专用管理域 (每实例一个)，以及多达 14 个由最终用户创建的虚拟基础设施工作负载域。

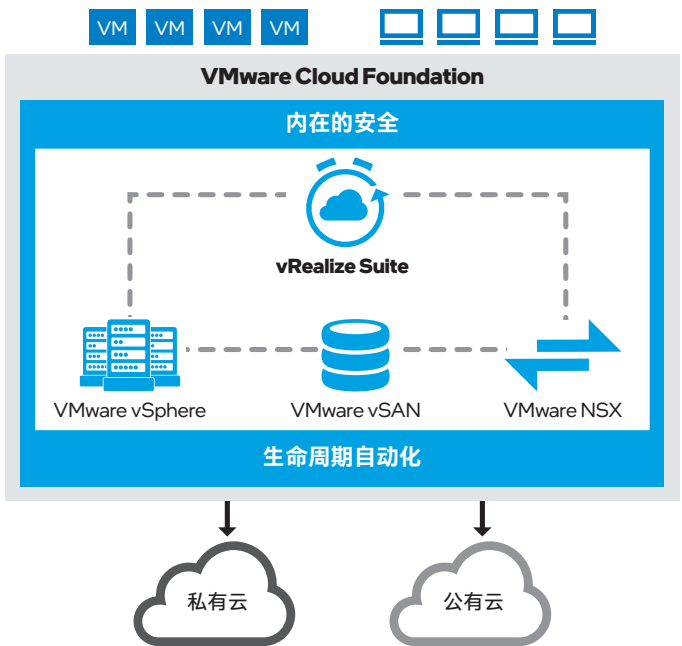


图 16. VMware Cloud Foundation 环境
来源: docs.vmware.com/en/VMware-Cloud-Foundation/3.9/com.vmware.vcf.ovdeploy.doc_39/GUID-7EBCC024-9604-4064-90A1-4851A78F7641.html

管理域

管理域是一个特殊用途的工作负载域，用于托管实例化、管理和监控 VMware Cloud Foundation 基础设施所需的基础设施组件。管理域是在 VMware Cloud Foundation 系统启动期间使用第一个机架上的 Cloud Builder 自动创建的。管理域包含 SDDC Manager、vCenter Server、NSX-T 管理集群和 vRealize Log Insight 等管理组件。管理域使用 vSAN 作为主存储，至少需要四个节点才能正常工作。当系统中添加更多机架时，管理域会自动集成附加组件。

工作负载域

工作负载域是私有云容量的逻辑分组，由 SDDC Manager 自动配置。工作负载域实行独立管理和补丁安装，有各自的计算、存储和网络资源可供使用。与工作负载域相关的所有任务都通过 SDDC Manager Web 界面执行，包括创建、扩展和删除工作负载域以及监控和管理物理基础设施。了解 VMware Cloud Foundation 的常见问题，请单击[这里](#)。

VMware vSAN

VMware vSAN 是与 VMware vSphere 完全集成的存储虚拟化软件，可将 vSphere 集群中的所有存储设备集成到一个共享数据池 (参见图 17)。使用 vSAN 后就不再需要外部的共享存储。

vSAN 集群有两种可能的配置:

- 混合 vSAN 集群使用两种类型的存储设备: 用于缓存层的闪存设备和用于容量层的磁碟驱动器。
- 全闪 vSAN 集群在缓存层和容量层都使用闪存设备。

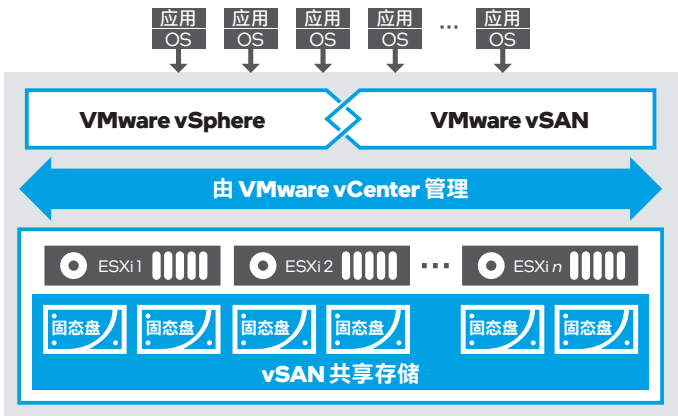


图 17. VMware vSAN 概览

VMware NSX

VMware NSX 是一个网络虚拟化解决方案，用于在虚拟化的数据中心创建软件定义网络。正如虚拟机是从物理服务器硬件中抽象出来的，虚拟网络（包括交换机、端口、路由器和防火墙等）也是在虚拟空间中构建的。虚拟网络的配置和管理独立于底层硬件。

VMware 提供两种 NSX 平台类型：NSX-V 和 NSX-T。NSX-V 仅适用于 vSphere 部署。它就是原始的 NSX 平台，已面世多年，并且始终绑定在单个 VMware vCenter Server 实例上。NSX-T 则面向众多虚拟化平台和多虚拟机管理程序环境，例如，KVM、Docker、Kubernetes、OpenStack 以及 AWS 原生工作负载等。NSX-T 部署不要求使用 vCenter Server，并且已经针对异构计算系统进行调整。NSX-T 包含 NSX-T 容器网络接口 (CNI) 插件，可用于为容器应用配置网络连接。

NSX 组件

VMware NSX 的主要组件包括 NSX Manager、NSX Controller 和 NSX Edge 网关：

- **NSX Manager** 是 NSX 中用于网络管理的集中式组件。该虚拟设备提供图形用户界面 (GUI) 和 RESTful API，用于创建、配置、编排和监控 NSX-T Data Center 组件。NSX Manager 是 NSX-T Data Center 生态系统的管理平面。
- **NSX Controller** 组成了一个分布式状态管理系统，用于覆盖传输隧道和控制虚拟网络，可作为虚拟机部署在 VMware ESXi 或 KVM 虚拟机管理程序上。NSX Controller 负责管理网络中所有的逻辑交换机，并处理有关虚拟机、主机、交换机和虚拟可扩展局域网 (VxLAN) 的信息。通常部署三个控制器节点即可确保一个 NSX Controller 发生故障时能具备充分的数据冗余。
- **NSX Edge** 是一项网关服务，为虚拟机提供对物理和虚拟网络的访问。NSX Edge 可以安装为分布式虚拟路由器或服务网关。它可以提供的服务包括：动态路由、防火墙、网络地址转换 (NAT)、DHCP、虚拟专用网络 (VPN)、负载均衡和高可用性。NSX Edge 可以连接到两个传输区域：一个用于覆盖网络，另一个用于与外部设备之间的南北向对等连接（参见图 18）。

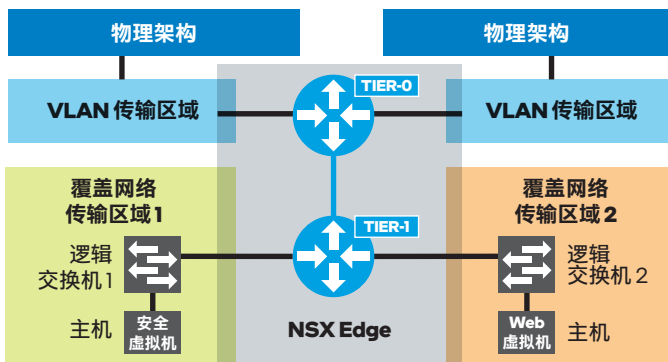


图 18. 使用 VMware NSX 的 VLAN 和覆盖网络传输区域

来源: docs.vmware.com/cn/VMware-NSX-T-Data-Center/2.3/com.vmware.nsx.install.doc/GUID-F47989B2-2B9D-4214-B3BA-5DDF66A1B0E6.html

这两个传输区域定义了 NSX Edge 上逻辑网络分布的限制。

- **覆盖网络传输区域**：任何加入 NSX-T Data Center 域的虚拟机的流量都可能需要能够访问外部设备或网络。通常这类流量被称为外部南北向流量。NSX Edge 节点负责解封从计算节点收到的覆盖网络流量，以及封装发送到计算节点的流量。
- **VLAN 传输区域**：除了封装或解封流量功能外，NSX Edge 节点还需要使用 VLAN 传输区域以提供连接到物理基础设施的上行链路连接。

NSX-V 要求使用 vSphere VDS，就像在 vSphere 中一样。NSX-V 不能使用标准虚拟交换机。NSX-T 假定您已经部署 NSX-T 分布式虚拟交换机 (N-VDS)。KVM 主机使用 Open vSwitch (OVS)，而 ESXi 主机使用 VMware vSwitch。

N-VDS 是执行流量传输的传输节点上的软件 NSX 组件，是传输节点数据平面的主要组件，用于转发流量，并拥有至少一个物理网卡。不同传输节点的每个 N-VDS 彼此独立，但是可以通过相同的命名把它们分成一组，从而实现集中管理。

NSX-V 和 NSX-T 均有可用的传输区域。传输区域定义了逻辑网络分布的限制。每个传输区域都与其 N-VDS 相连。NSX-T 的传输区域不与集群相连。由于采用 GENEVE 封装，VMware NSX-T 有两种类型的传输区域：覆盖网络和 VLAN。而对于 VMware NSX-V，传输区域仅定义 VxLAN 的分布限制。

Tanzu Kubernetes Grid Integrated

Tanzu Kubernetes Grid Integrated (TKGi) 解决方案可帮助多云企业和服务提供商实施 Kubernetes（参见图 19）。VMware Cloud Foundation 支持 TKGi 在 NSX-T 工作负载域上的自动化部署。TKGi 可通过开发和部署 (Day 1) 以及出厂和交付 (Day 2) 阶段的运维支持简化 Kubernetes 集群部署，并能够管理从应用层到基础设施层的容器部署。部署的 Kubernetes 集群以原生形式提供，不含插件或专有扩展。您可以使用命令行界面配置整个 Kubernetes 集群，并使用原生 Kubernetes 命令行界面 (kubectl) 在集群上运行基于容器的工作负载。有关 TKGi 的更多信息，请单击[这里](#)。

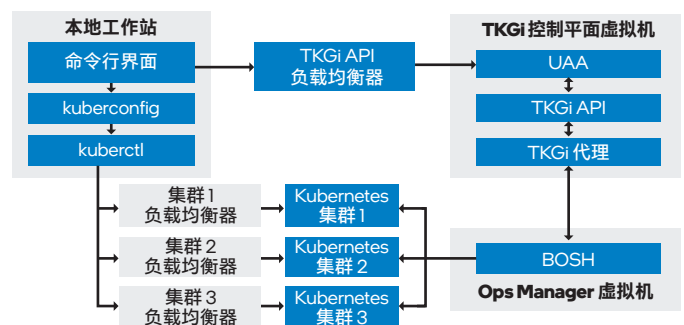


图 19. TKGi 控制平面

来源: docs.pivotal.io/pks/1-6/control-plane.html

集成 TKGi、VMware Cloud Foundation、NSX-T 和 vSAN

VMware Cloud Foundation 整合了计算、存储、网络、安全和云管理服务，为运行企业工作负载和容器化应用创建了一个理想的平台。vSAN 可以按需灵活定义策略，轻松简化容器的存储管理。开发人员可以将底层存储基础设施的复杂性抽象出来，从而以代码形式使用存储。借助 NSX-T，最终用户不再需要了解底层网络架构。NSX-T 可以自动创建 TKGi 使用的负载均衡器、路由器和交换机（参见图 20）。

基于面向 Kubernetes 集群中容器的 VMware Cloud Foundation 网络紧密集成 TKGi、NSX-T 和 vSAN（参见图 21），不仅有助于轻松管理临时和持久存储，还能获得 vSAN 的可用性和数据服务功能。此外，vSphere 高可用性和 VMware vSphere 容错机制也可以保护虚拟机免受物理服务器故障的影响。这些技术的结合将 TKGi on VMware Cloud Foundation 打造成了一个完整的解决方案，非常适合 Kubernetes 管理员和开发人员使用。

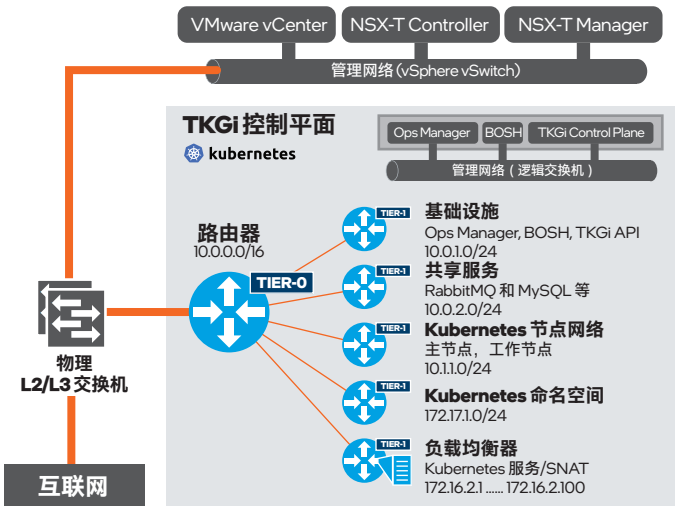


图 20. 使用 NSX-T 虚拟网络路由器的 TKGi
来源: docs.vmware.com/en/VMware-Enterprise-PKS/1.6/vmware-enterprise-pks-16/GUID-nsxt-multi-pks.html

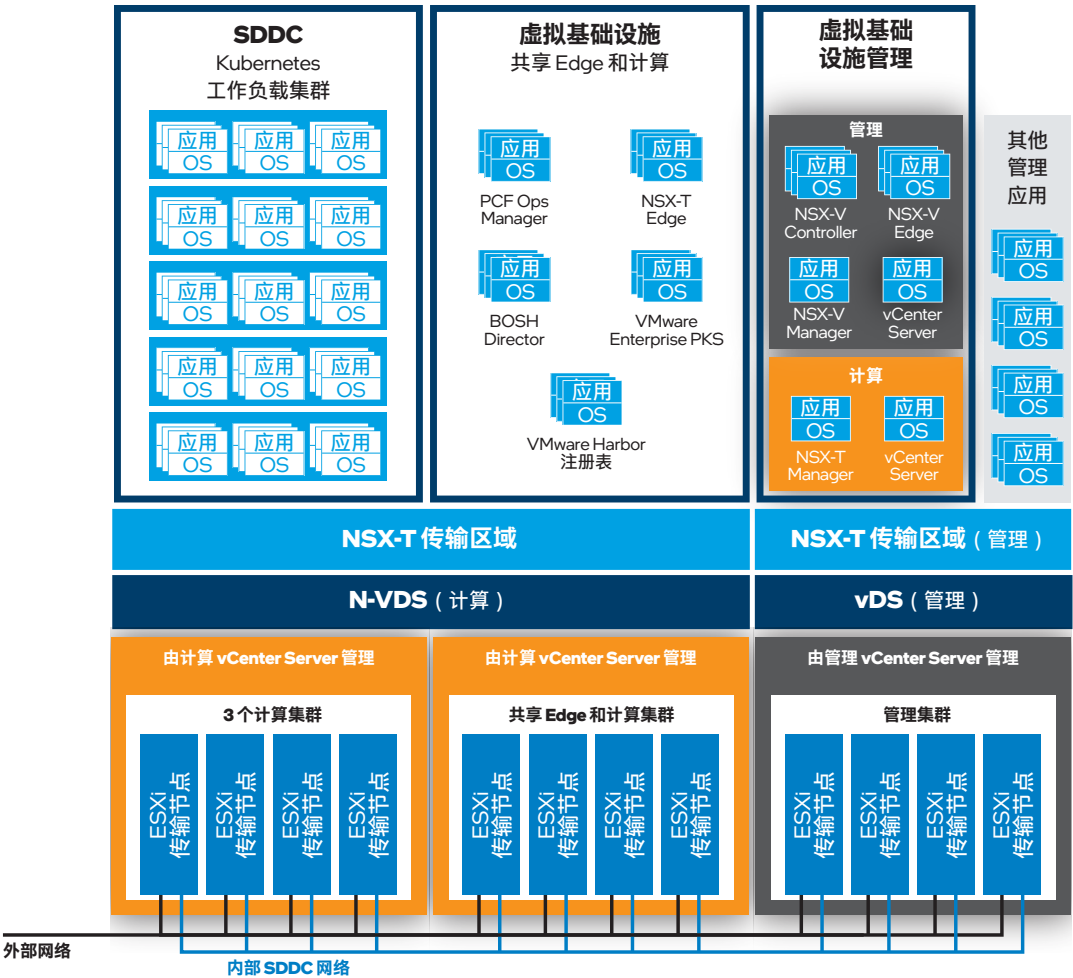


图 21. 集成 TKGi 并使用 VMware NSX-T 的 VMware Cloud Foundation 生态系统
来源: docs.vmware.com/en/VMware-Validated-Design/5.1/sddc-architecture-and-design-for-vmware-enterprise-pks-with-vmware-nsx-t-workload-domains/GUID-8F4D6F40-8126-4C41-952D-192A45AF5AF3.html

VMware Cloud on AWS

VMware Cloud on AWS 是一个完整的公有云上的 VMware 解决方案（参见图 22）。它以服务的形式出售和运营，包含与本地环境相同的组件：vSphere、vSAN、NSX-T 和 vCenter Server。它能够快速地将一般的 VMware 环境直接扩展并迁移到 AWS 公有云、提供保护并且无缝集成 Kubernetes 部署。借助额外的工具和插件（例如 HCX 和混合链接模式），它能够提提供虚拟机上云/下云的方法。服务本身具有两个不同的预配置区域：一个用于管理，另一个供客户使用。VMware 负责管理部分，客户控制操作部分。用户对管理资源和设置的访问权限相当有限，但可以对计算资源池中的工作负载进行管理。

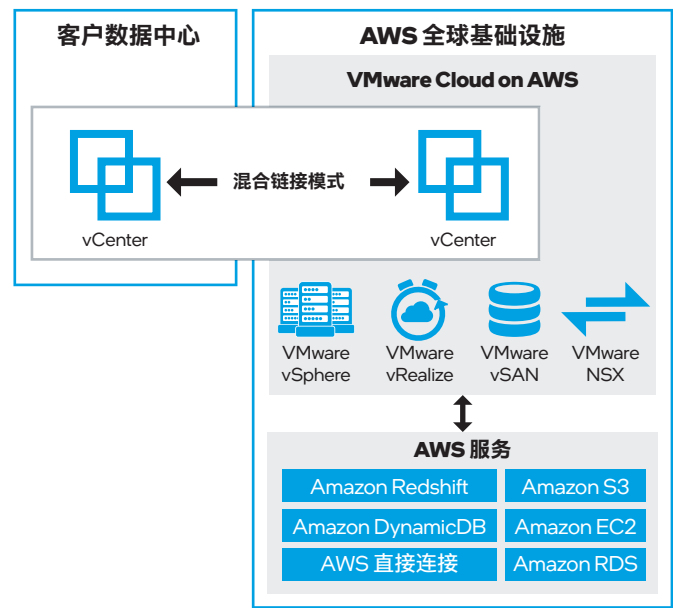


图 22. VMware Cloud on AWS 的组件

来源: youtube.com/watch?v=OI7rNfkZT2c

网络层面上，有两个网关提供与 VMware Cloud Foundation 的连接。通过管理网关 (MGW)，用户能够连接到使用专用管理子网和受限端口的管理层（包含 vCenter、ESXi 主机、NSX、SRM 和 HCX 管理器）。而计算网关 (CGW) 则为工作负载虚拟机网络流量流进和流出 VMware Cloud 提供了入口和出口。分布式防火墙功能可以过滤 VMware Cloud 中不同分段上的虚拟机之间的流量。计算网关或分布式防火墙不受限制，用户可以自行配置防火墙规则。管理网关和计算网关使用单独的 VMware NSX Edge。在部署阶段，用户可以就两种基于英特尔® 架构的 Amazon EC2 实例（即 i3.metal 或 r5.metal）、与 AWS 原生环境的集成以及连接选项做出选择。更多有关实例的详细信息，请单击[这里](#)。

更多有关 VMware Cloud on AWS 的详细信息，请参阅 [VMware Cloud on AWS 深度技术研究](#) 演示。

环境配置和部署

在 VMware ESXi 上安装和配置英特尔® 傲腾™ 持久内存模组

要在 ESXi 系统上使用英特尔® 傲腾™ 持久内存模组，必须在每台服务器上安装和配置该模组。第一步是将模组安装到主板上的 DIMM 插槽中。具体可实现的配置存在一些限制，您可参阅[平台手册](#)了解详细的配置规则和指南。其中一个限制是：为 CPU1 安装的 DIMM 类型和数量配置必须与 CPU2 匹配。您可以在平台手册中找到所有可能的插槽配置详细图表。

英特尔® 傲腾™ 持久内存模组安装完毕后，需要选择内存模式或 App Direct 模式。更多详情，请参阅[英特尔® 傲腾™ 持久内存 - BIOS 设置](#)。对于 ESXi，唯一的要求是为区域创建目标配置。如果您打算使用 App Direct 模式，请将“内存模式 [%]”选项设置为 0（您仍需要创建该区域）。如果您希望使用内存模式来获得额外内存，请将“内存模式 [%]”选项的值更改为 100。每次创建新的目标配置后都需要重新启动服务器。您无需手动创建命名空间；如有需要，ESXi 会在启动过程中自动创建命名空间。然后，系统就可以投入使用了。

重要：如果您想从内存模式更改为 App Direct 模式，或从 App Direct 模式改为内存模式，必须在更改 BIOS 之后，删除 ESXi 创建的命名空间。请遵循[删除 VMware Host Client 中的命名空间](#)的说明。

为英特尔® 傲腾™ 持久内存准备虚拟机

如果您计划使用 App Direct 模式，在配置虚拟机时需要完成额外步骤。对于虚拟机上的任何客户机操作系统，步骤基本相似；您可阅读[使用英特尔® 傲腾™ 持久内存增强 VMware 虚拟机](#)指南，大致了解相关步骤。在 Linux 机器上，您将使用 ndctl 工具，而 Windows Server 2019 则有一组集成命令可用于达到相同的状态（有关如何在 Windows Server 上配置 DAX，请参阅[了解和部署持久内存](#)）。

重要：如果您想使用 DAX 模式，虚拟机上的客户机操作系统必须支持持久内存。

如果客户机操作系统不支持 DAX，您仍可以在虚拟机上使用 App Direct 模式，但只能用作普通文件系统；不过，您仍将获得性能提升。

如果您使用内存模式，则无需额外的步骤来配置虚拟机。虚拟机会将英特尔® 傲腾™ 持久内存用作一般的 DRAM 模组。

环境置备

如“解决方案概述”一节所述，完整的环境包含三个主要产品：VMware Cloud Foundation、VMware NSX-T 和 TKGi。下面为您介绍如何置备这些组件。

软硬件要求

在部署 VMware Cloud Foundation 平台之前，必须满足多项要求。[VMware Cloud Foundation 规划和准备指南](#)中列出了完整的要求详情。

网络（巨型帧和 802.1Q 标记）、网络池、VLAN 和 IP 池、主机名和 IP 地址都有相应的具体要求。您应在熟悉上述规划和准备指南之后再采取后续步骤。整体设置在很大程度上依赖多个 VLAN。您可以根据自身的需求为每个域设置一组自己的独立 VLAN，用于管理、VMware vMotion、vSAN、VxLAN（NSX VTEP）和上行链路。更多详细信息，请参阅 VMware 的 [VLAN ID 和 IP 子网](#) 文档。

重要：使用 GENEVE 封装时，巨型帧的最大传输单元（MTU）值必须至少为 1,700 字节。这是因为 GENEVE 报头有一个额外的长度可变的元数据字段。如果不满足此要求，覆盖网络将无法正常工作。

VMware Cloud Foundation 管理域部署

VMware Cloud Foundation 部署过程包含多个步骤。但在部署之前，您需要先获得所有必要的硬件组件，将它们安装到机架中，准备好电力和散热设施，并建立连接到数据中心基础设施的上行链路。接着，您就可以开始部署流程。首先部署 Cloud Builder 虚拟机，用于管理初始的管理域配置过程。如果您尚未安装 ESXi 操作系统，Cloud Builder 虚拟机也可用于在节点上安装 ESXi 操作系统，但这一步由用户自行选择。然后您需要下载并填写部署参数表，最后开始 VMware Cloud Foundation 启动流程。

有关 VMware Cloud Foundation 的完整部署过程，请参阅 [VMware Cloud Foundation 架构和部署指南 - VMware Cloud Foundation 3.9](#) 文档中的“部署 Cloud Foundation”一章。下文简单介绍四个步骤。

第 1 步：部署 Cloud Builder 虚拟机

Cloud Builder 虚拟机可用于部署 VMware Cloud Foundation，它还包括用于生成 ESXi 服务器映像的 VMware Imaging Appliance（VIA）。您也可以不使用 VIA，而是手动在所有节点上安装 ESXi。更多详细信息，请参阅 [VMware Cloud Foundation 规划和准备指南](#) 中的“软件要求”一章。

有关 Cloud Builder 虚拟机的详细部署步骤，请参阅 VMware 的 [部署 Cloud Builder 虚拟机](#) 文档。

第 2 步：在 VMware Cloud Foundation 服务器上安装 ESXi 软件

使用 Cloud Builder 虚拟机创建 ESXi 服务器映像（通过 VIA 完成）为可选步骤。如果您的服务器已安装受支持的 ESXi 版本，则您无需使用 VIA。您也可以在每台机器上手动安装 ESXi。使用 VIA 的优势在于它不仅可以安装 ESXi，还可以部署额外的 vSphere 捆绑安装包（VIB），并在所有机器上配置标准密码。

有关如何准备主机和设置 VIA 以安装 ESXi 的详细信息，请参阅 [Cloud Foundation 指南](#) 中的“在 Cloud Foundation 服务器上安装 ESXi 软件”一章。

此外，请务必安装您的服务器要求的 VIB 或服务器所需的自定义 VIB。大多数情况下，这些 VIB 会用作网卡或固态盘的特定驱动程序。如果您决定使用 VIA，可以通过“**Bundle → Modify VIBs**”的菜单操作添加任何所需的 VIB，这样 VIB 就会随 ESXi 一起自动进行预安装。如果您不使用 VIA，则需要手动安装 VIB。在本参考架构中，我们添加了以下 VIB：

- **英特尔® 网络适配器的网卡驱动程序和更新工具：**如您需下载面向以太网控制器 X710、XL710、XXV710 和 X722 系列的 VMware ESXi 6.7 i40en 1.9.5 网卡驱动程序，请单击[这里](#)。
- **面向 NVMe 的英特尔® VMD 驱动程序：**如您需下载面向英特尔的 NVMe 驱动程序 VMware ESXi 6.7 intel-nvme-vmd 1.8.0.1001，请单击[这里](#)。
- **英特尔® 固态硬盘数据中心驱动管理工具：**如您需下载 Intel SSD_DCT_3.0.24_ESXi，请单击[这里](#)。

第 3 步：下载并填写 VMware Cloud Foundation 部署参数表

该参数表是一个电子表格文件，用于收集所有环境信息。您需要从 vmware.com 下载这个独立文件。完成所有必填字段（包括 VLAN 列表、网络地址、上行链路、密码和许可证）后，您可在 VMware Cloud Foundation 启动过程中导入此文件。

您可以在《VMware Cloud Foundation 架构和部署指南》中找到完整的“部署参数表”文档，具体请参阅[下载并填写部署参数表](#)。该文档详细描述了参数表的所有页签和字段。

重要：请确保您的密码满足参数表中的密码复杂度条件。如果密码不符合规则，部署将会失败。

第 4 步：启动 VMware Cloud Foundation

如果您已在所有管理节点上安装了 ESXi，添加了所需的自定义 VIB 并完成了部署参数表的填写，您就可以启动 VMware Cloud Foundation 了。

启动过程中会部署 SDDC Manager、用于管理域的 vCenter Server、NSX Manager 和 Controller、平台服务控制器 (PSC)、vSAN 和 vRealize Log Insight (即, VMware Cloud Foundation 的完整管理域)。这个过程大约需要两个小时。

启动过程完成后, 您会看到一条通知, 其中包含新 SDDC Manager 的 Web 界面链接。您可以通过标准 Web 浏览器访问 SDDC Manager 界面。

有关 VMware Cloud Foundation 启动流程的完整描述, 请参阅 VMware 的[开始 Cloud Foundation 启动流程](#)文档。

至此, 管理工作负载域已经创建完成, 其中包含管理基础设施所需的所有组件。请不要在此管理集群中部署用户应用。相反, 您可以创建一个或多个工作负载域, 在其中部署已预安装和配置了 vSAN 和 NSX 的一个或多个独立 vSphere 集群, 并为每个工作负载域创建额外的专用 vCenter Server 实例。vCenter Server 实例 (每个工作负载域一个实例) 和 NSX-T 管理集群将在后续步骤中部署到管理域。

重要: 部署 SDDC Manager 和管理域后, 您必须配置双重身份验证。在使用 SDDC Manager 执行特定任务时 (如管理整个平台的密码和配置 NSX Manager 备份), 双重身份验证是必要的前提条件。有关详细信息, 请参阅 VMware 文档[“配置双重身份验证”](#)。

VMware Cloud Foundation 工作负载域部署

第 1 步: 用 NSX-T 部署工作负载域的前提条件

由于本参考架构使用 NSX-T (而非 NSX-V) 进行网络连接, 因此必须将 NSX-T 捆绑安装包下载到 SDDC Manager:

1. 如需浏览并将捆绑安装包下载到 SDDC Manager, 您可选择“**Administration → Repository Settings**”, 然后输入您的 VMware 帐户凭证信息。
2. 搜索 NSX-T Manager 捆绑安装包 (本解决方案使用的安装包名称为 NSX_T_MANAGER 2.5.0.0.0-14663974), 您可以在“**Repository → Bundles**”中下载捆绑安装包。

SDDC Manager 上的 NSX-T Manager 捆绑安装包是必需组件, 因为您创建第一个使用 NSX-T 的工作负载域时, SDDC Manager 会在管理集群中部署一系列额外的虚拟机, 包括一组三个的 NSX-T Manager 虚拟机。所有后续 NSX-T 工作负载域都将共享此 NSX-T Manager 集群。只有在还没有其他 NSX-T 工作负载域时, 您才需要执行此步骤。

您还必须为 VxLAN (主机 VTEP) 准备专用 VLAN 并启用 DHCP。此外, 所有 VLAN 和接口的 MTU 值必须至少为 1,700 (建议为 9,000)。如需详细了解所有前提条件, 请单击[这里](#)。

为了进一步部署 Edge 虚拟机, 您还需要为 Edge VTEP 准备额外的 VLAN, 并为上行链路再准备两个 VLAN。主机 VTEP 和 Edge VTEP 的 VLAN 之间必须能够进行路由。

第 2 步: 创建工作负载域

SDDC Manager 会控制和编排 VMware Cloud Foundation 上工作负载域的创建, 并负责安装和配置所有必需组件 (包括 vCenter Server、vSAN 和 NSX-T)。最终用户必须使用 VI 配置向导来部署新的工作负载域。如果这是环境中第一个使用 NSX-T 的工作负载域, 向导将要求提供其他信息 (例如 NSX-T Manager 虚拟机的 DNS 名称), 因为除了常规工作负载域外, 还要创建一个 NSX-T 底板。

有关创建工作负载域的完整步骤, 请参阅 [VMware Cloud Foundation 操作和管理指南](#)文档中的“[启动 VI 配置向导](#)”一章。

创建一个工作负载域所需的时间取决于服务器配置和需要创建的基础设施大小。在本示例环境中, 配置完整的包含四台服务器的工作负载域基础设施需要约 1 小时 40 分钟。这意味着, 根据系统和网络配置的不同, 置备这样一套环境可能只需要短短两个小时, 而在 VMware Cloud Foundation 中还没有 SDDC Manager 提供自动化功能之前, 这个过程通常需要几天的时间。

此外, 由于整个流程均为自动化实施, 手动安装中常见的配置错误风险也可大大降低。这种配置错误可能会导致基础设施配置出现严重问题或更多延迟。

VI 配置向导运行完成后, 将生成一个具有以下配置和网络结构的新集群:

- 每台主机两个传输区域 (一个用于 VLAN, 另一个用于覆盖网络)
- 网络 I/O 控制 (NIOC) 和主机上行链路配置文件
- 四台逻辑交换机 (管理、vSAN、vMotion 和覆盖网络)
- 一些空闲的 VDS 和端口组

重要: 请勿编辑或删除上述任何配置。

在新创建的工作负载域中添加 NSX-T Edge 主机

工作负载域部署完成后，接着就要创建和配置 NSX-T Edge 虚拟机。为了实现南北流量传输，需要使用 NSX Edge 主机来启用覆盖网络虚拟基础设施和公共网络。工作负载域创建过程中不会自动部署 NSX Edge 主机。

有关如何在工作负载域中部署 NSX Edge 虚拟机，请参阅在 [Cloud Foundation 中部署和配置 NSX Edge](#) 文档。您必须为您创建的每个工作负载域执行此过程。执行过程中，您可能会注意到某些对象（如 vlan-tz 和覆盖网络传输区域）在第一个工作负载域创建过程中已自动创建完成。请勿编辑、删除或重新创建这些对象，您仅需创建缺失的对象（例如传统上行链路传输区域、缺失的分段和 IP 池等）。

重要：除了本文引用的 VMware Cloud Foundation 3.9 文档，[VMware 的 VMware 验证设计文档](#)也提供了有关如何在 VMware 环境中建立 NSX 的指南。根据后一份文档，您必须创建一个额外的网络分段才能使 NSX-T Edge 虚拟机正常工作；但是，VMware Cloud Foundation 3.9 文档中并未提供该分段相关信息（完整分段列表，请[在此查看](#)）。当您看到文档的“为系统、上行链路和覆盖网络流量创建 NSX-T 分段”一节时，请使用下方表 5 提供的分段列表，而不是上述文档中的表格。

表 5. 所需分段的完整列表

分段名称	传输区域	VLAN	类型
nvds01-uplink01	sfo01-esxi-vlan	0-4094	Trunk
nvds01-uplink02	sfo01-esxi-vlan	0-4094	Trunk
sfo01-w-uplink01	sfo01-w-uplink01	1647	Access
sfo01-w-uplink02	sfo01-w-uplink02	1648	Access
sfo01-w-overlay	sfo01-esxi-vlan	0-4094	Trunk

VMware Cloud Foundation 3.9 文档不包含表 5 中的最后一行（黄色高亮显示部分），但该文档中的下一步骤使用了该分段（Edge 设备中的一个接口需要连接到该分段）。如果没有这一行，您就无法参照文档操作，因此，请确保添加该行内容。与缺失的这一分段类似，您还必须创建覆盖网络配置文件（VMware Cloud Foundation 3.9 文档中也缺失了相关内容）。请参阅验证设计文档的“[创建上行链路配置文件](#)”一节（见该节第 4 步中的表格，查找 sfo01-w-overlay-profile），以了解所需覆盖网络配置文件的相关值。

您可以通过以下两种方式之一部署 Edge 设备：您可参照文档的指导，使用开放虚拟化格式 (OVF) 模板进行部署，或者在

NSX-T Manager 的 Web 用户界面上进行部署。如果使用 Web 用户界面，您就无法通过向导将正确的分段与虚拟机接口相连 - 因为您只能看到管理域中的网络。您只需保持默认布局不变，等待虚拟机出现在 vCenter 中，就可以通过编辑虚拟机设置看到所有必要的分段并对网络进行更改了。

Tanzu Kubernetes Grid Integrated 部署

现在您已拥有一个由 NSX-T 和 NSX-T Edge 主机组成的工作负载域，接下来就可以开始安装 Tanzu Kubernetes Grid Integrated (TKGi)。

该部署过程大部分是自动化完成的。您可参照[部署企业 PKS 解决方案](#)文档中的步骤核对前提条件，并配置其他子网、主机名和 IP 地址。请注意，您必须获取所有 TKGi 组件（BOSH、Ops Manager 和 Harbor 等）的现有 DNS 条目。验证这些条目是否与部署过程中 TKGi 分配给其虚拟机的 IP 地址一致。如果不一致，部署将会失败。

您可以为每个工作负载域部署一个 TKGi 实例，在工作负载域内部署 TKGi 并不会限制域本身，只要集群上的资源可用，它仍可用作普通的工作负载域。

VMware Cloud on AWS 配置

本节将介绍启动 VMware Cloud on AWS 并将其连接到本地环境所需的组件和步骤。

创建 SDDC Manager

启动云环境的第一步是从 VMware Cloud 控制台部署 SDDC。这个过程很简单，只需要选择服务所在的 AWS 区域，然后选择 SDDC 的部署选项（SDDC 中有一个或多个主机时，可以选择延伸集群）、主机类型和名称。此外，您必须（在 14 天内）将新创建的 SDDC 关联到 AWS 帐户。有关完整流程的详细内容，请参阅[从 VMC 控制台部署 SDDC](#)文档。整个流程结束后，具备计算集群的完整 vCenter 环境便已准备就绪。部署完成后，请务必在 VMware Cloud 控制台的 SDDC 设置中添加必要的防火墙规则。默认情况下，所有流量都会被拦截，如果您不更改这些规则，将无法使用 SDDC 环境。

VPN 配置

要从本地服务安全地访问 SDDC，需要建立专用连接。为此，您可建立 VPN 连接或使用 AWS 直接连接 (DX)。在本参考架构中，我们配置了站点到站点 IPsec VPN。如果从 SDDC 端进行配置，这一过程相对比较容易，但本地端往往需要详细的规划和预配置。每个环境都是独特的，需要额外的配置来准备隧道端点、路由和防火墙规则。本地隧道端点类型定义了建立隧道所需的确切设置，而且隧道两端的设置必须一致。本例使用了基于策略的 VPN，但您可以根据自身需求和环境改用基于路由的 VPN 和边界网关协议 (BGP)。最终用户也可以不通过 VPN 连接到 VMware Cloud on AWS，但在使用部分混合云的先进功能时，这样做的安全性不如使用 VPN 或 DX。

有关 VPN 逐步配置过程的信息，请参阅“[VMware Cloud on AWS：使用 VPN 建立连接](#)”文章和 VMware Docs 上的“[在 SDDC 和内部部署数据中心之间配置 VPN 连接](#)”。

HCX 部署

VMware HCX 为混合云环境提供了额外的特性和功能，包括站点互连，vMotion 和批量虚拟机迁移支持、网络扩展 (L2 相邻分段)、业务持续性和容灾保护、虚拟机复制和 WAN 优化等。VMware HCX 是一种站点到站点架构，其源环境和目标环境有显著区别 (两者有各自的特定安装程序)。通常，HCX Cloud

由云服务提供商部署在公有云中，HCX Connector 则安装在本地。无论哪种类型，HCX 总是靠近 vCenter Server 部署在管理区域。生成的 HCX Manager 用于管理 VMware HCX (作为新图标和菜单选项添加到 vCenter Server 中)。

HCX 由源站点和目标站点的虚拟管理机组成，其中包含多达五种互连服务设备 (具体取决于所选的功能和 HCX 许可)。这些服务在源站点作为虚拟设备启用和部署，在目标站点则与源站点对应。一般部署工作流程如图 23 所示。有关如何部署 HCX 与互连服务网络的详细说明，请参阅 [VMware HCX 用户指南](#)。

启用混合链接模式

出于管理便捷性，您可以配置混合链接模式。该模式支持单点登录到本地和云端 vCenter 进行管理操作。这样就可以把管理集中到一处，通过一台额外的本地虚拟机设备实现集中管理。在该设备中，您可以看到整个基础设施，就如同连接到了一个平台服务控制器。有关详细的部署信息，请参阅 VMware Docs 上的“[配置混合链接模式](#)”章节。

重要： 配置混合链接模式之前必须先配置 VPN，因为云端 vCenter 只能通过内部 (非公共) 云 IP 地址访问，因此要求必须配置 VPN。



图 23. VMware HCX 部署工作流程
来源: docs.vmware.com/en/VMware-HCX/services/hcx-user-guide.pdf

总结

在日益数字化的世界，对高性能数据分析和人工智能的需求与日俱增，企业寻求能够在本地或公有云中运行传统数据分析以及人工智能应用的灵活解决方案。VMware 混合云平台将出色的英特尔® 硬件（包括英特尔® 傲腾™ 持久内存）和 VMware 虚拟化软件相结合。借助这一可随时部署的端到端解决方案，企业既能运行传统的数据分析工作负载，又能运行面向未来的人工智能和机器学习工作负载。

同时，您要了解到，数据仓库是时延和 I/O 敏感型工作负载，而数据分析则是 CPU 和内存密集型工作负载。本参考架构已经过验证，在要求严苛的客户工作负载场景中能够达到预期关键性能指标。具体结果如下：

数据仓库

- 增强配置每分钟事务处理量高达基础配置的 3.34 倍。
- 对于基础集群和增强集群，至少 98% 的 Microsoft SQL Server 请求在 CPU 内的时间小于 5 毫秒。
- 除要求最为严苛的场景外，增强配置始终达到“80% 的请求在 CPU 内的总时间小于 20 毫秒”的 SLA 值要求。即使在要求最为严苛的场景中，性能也只有小幅下降。

数据分析和人工智能

- 我们观察到当人工智能推理工作负载有更多资源可用时，吞吐量呈线性提升，证明了出色的可扩展性。
- 使用深度学习参考堆栈的优化版 TensorFlow 后，无论是基础配置还是增强配置，帧率均可达到标准版 TensorFlow 的 2.2 至 3 倍。

找到适合您的企业或机构的解决方案。请联系您的英特尔代表或访问面向 VMware vSAN 的英特尔® 精选解决方案。

了解更多信息

以下资源可能对您有所帮助：

- [面向云的英特尔® 数据中心模块 - vSAN ReadyNodes](#)
- [英特尔® 傲腾™ 持久内存产品主页](#)
- [英特尔® 傲腾™ 持久内存 - 虚拟化性能](#)
- [深度学习参考堆栈](#)
- [第二代英特尔® 至强® 可扩展处理器](#)
- [面向 VMware vSAN ReadyNodes 的英特尔® 精选解决方案](#)
- [英特尔® 傲腾™ 固态硬盘](#)
- [英特尔® 固态硬盘数据中心家族](#)
- [英特尔® 分发版 OpenVINO™ 工具包](#)
- [英特尔® 深度学习加速技术](#)
- [面向英特尔® 架构的框架优化](#)
 - [VMware vSAN](#)
 - [VMware Cloud Foundation](#)
 - [VMware Cloud on AWS](#)

附录 A：解决方案功能验证和基准测试

本节介绍如何重现功能验证实验，包括一些必需组件和软件的安装步骤、配置说明和基准测试执行脚本。

配置 Oracle 19c 以在内存模式下使用英特尔® 傲腾™ 持久内存

要在 Oracle 数据库中使用内存模式下的英特尔® 傲腾™ 持久内存，您需要把 VMware 物理服务器上的持久内存配置为使用内存模式。然后，在为 Oracle 数据库创建虚拟机时，您可以为虚拟机选择比从前更大的可用内存量。您可以为虚拟机分配 1.5 TB、3 TB 甚至高达 6 TB 的内存（取决于 CPU 型号）。

配置 Oracle 19c 以在 App Direct 模式下使用英特尔® 傲腾™ 持久内存ⁱⁱⁱ

要在 Oracle 数据库中使用 App Direct 模式下的英特尔® 傲腾™ 持久内存，您必须遵循此[文档](#)，将 VMware 物理服务器上的持久内存配置为使用 App Direct 模式。这将会把持久内存配置为支持 DAX 的文件系统。然后，您需要将这些磁盘指定为 Oracle Grid 配置的专用磁盘。这些磁盘将用于 +DATA 和 +REDO。具体步骤如下：

1. 根据以下代码，在您的系统上准备磁盘。

```
[root@oracledb ~]# parted /dev/pmem0
GNU Parted 3.2
Using /dev/pmem0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Error: /dev/pmem0: unrecognised disk label
Model: NVDIMM Device (pmem)
Disk /dev/pmem0: 889GB
Sector size (logical/physical): 512B/4096B
Partition Table: unknown
Disk Flags:
(parted) mklabel gpt
(parted) mkpart primary 2048s 100%
(parted) align-check opt 1
1 aligned
(parted) print
Model: NVDIMM Device (pmem)
Disk /dev/pmem0: 889GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number Start End Size File system Name Flags
1 1049kB 889GB 889GB primary

(parted) quit
Information: You may need to update /etc/fstab.
```

```
[root@oracledb ~]# parted /dev/pmem1
GNU Parted 3.2
Using /dev/pmem1
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Error: /dev/pmem1: unrecognised disk label
Model: NVDIMM Device (pmem)
Disk /dev/pmem1: 889GB
Sector size (logical/physical): 512B/4096B
Partition Table: unknown
Disk Flags:
(parted) mklabel gpt
(parted) mkpart primary 2048s 100%
(parted) align-check opt 1
1 aligned
(parted) print
Model: NVDIMM Device (pmem)
Disk /dev/pmem1: 889GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number Start End Size File system Name Flags
1 1049kB 889GB 889GB primary

(parted) quit
Information: You may need to update /etc/fstab.
```

2. 准备好磁盘后，运行以下命令将这些磁盘的所有权更改为您的 Oracle 帐户和群组：

```
chown oracle:dba /dev/pmem0
chown oracle:dba /dev/pmem1
```

3. 现在，[下载 Oracle 软件归档](#)。运行以下命令下载归档（将 URL 更改为从前一个站点获得的网址）：

```
curl -o /tmp/LINUX.X64_193000_grid_home.zip https://download.oracle.com/otn/linux/oracle19c/190000/LINUX.X64_193000_grid_home.zip
curl -o /tmp/LINUX.X64_193000_db_home.zip https://download.oracle.com/otn/linux/oracle19c/190000/LINUX.X64_193000_db_home.zip
```

4. 运行以下命令安装 Oracle Grid，从而使用自动存储管理 (ASM) 来管理硬盘：

```
su - grid
cd /u01/app/grid/product/19.3/grid/
unzip -q /tmp/LINUX.X64_193000_grid_home.zip

su - root
cd /u01/app/grid/product/19.3/grid/cv/rpm
CVUQDISK_GRP=dba; export CVUQDISK_GRP
rpm -iv cvuqdisk-1.0.10-1.rpm

su - grid
cd /u01/app/grid/product/19.3/grid/
./gridSetup.sh
```

ⁱⁱⁱ 注：截至本文发布时的所有版本 Oracle 数据库在以下说明列出的配置中可能发生损坏，详见 [My Oracle Support 说明：数据库服务器持久内存中的文件系统和设备可能会导致数据库损坏](#)（文档 ID: 2608116.1）。然而，该说明中描述的问题不适用于内存模式下的持久内存。更多详情，请单击[这里](#)。

- 在 Oracle Grid 的安装窗口中，选择“**Configure Oracle Grid Infrastructure for Standalone Server**”，单击“**Next**”，并按以下步骤操作：

- 在下一个窗口中，选择您准备好的一个硬盘，然后将“**Redundancy**”更改为“**External**”。这样会禁用冗余功能，但能提高数据写入效率。请注意，此方法安全性较低，只能用于测试，一般不建议使用。
- 单击“**Next**”，然后在下一个窗口中输入帐户密码。DBA 用户组选择“**Oracle ASM Operator Group**”。“安装位置 (Installation Location)”和“创建库存 (Create Inventory)”步骤将读取 ENV。
- 单击“**Next**”，按照配置向导的指导操作，然后单击“**Install**”。
- Oracle Grid 成功安装完成后，运行以下命令检查配置（您应能看到 DATA 磁盘组）：

```
[grid@server ~]$ asmcmd
ASMCMD> ls
DATA/
ASMCMD> quit
```

- 创建一个 REDO 磁盘组。运行以下命令启动配置向导：

```
[grid@hammer-server ~]$ asmca
```

- 在配置向导中：

- 双击“**Disk Groups**”打开磁盘组列表。在窗口底部，单击“**Create**”以添加一个 REDO 磁盘组。
- 选择您准备好的磁盘，将“**Redundancy**”设置为“**External**”。这样会禁用冗余功能，但能提高数据写入效率。请注意，此方法安全性较低，只能用于测试，一般不建议使用。
- 单击“**OK**”接受更改。您会看到两个磁盘组：DATA 和 REDO。

下一步是安装 Oracle 数据库。请按照 Oracle [文档](#)中的说明操作。

使用深度学习参考堆栈容器运行 TensorFlow 基准测试（再现推理基准测试）

首先，您需要使用安全外壳协议 (SSH) 连接到您 VMware 环境中的一个节点或虚拟机。然后按照以下步骤操作：

- 安装 [Docker](#) 和 [Helm 3](#)。
- Docker 就绪后，准备一个包含以下内容的 Docker 文件 (dlrs.Dockerfile)：

```
FROM clearlinux/stacks-dlrs-mkl:v0.5.0

RUN mkdir /tf && cd /tf
RUN swupd clean && swupd bundle-add wget git devpkg-gperf tools sysadmin-basic
RUN git clone --depth 1 https://github.com/tensorflow/models.git /tf/tf_models
RUN git clone -b v1.5.0 --depth 1 https://github.com/IntelAI/models.git /tf/intel-models
RUN wget -q -P /tf https://storage.googleapis.com/intel-optimized-tensorflow/models/v1_5/resnet50v1_5_int8_pretrained_model.pb
RUN wget -q -P /tf https://zenodo.org/record/2535873/files/resnet50_v1.pb
RUN wget -q -P /tf https://storage.googleapis.com/intel-optimized-tensorflow/models/v1_5/inceptionv3_int8_pretrained_model.pb
RUN wget -q -P /tf https://storage.googleapis.com/intel-optimized-tensorflow/models/v1_5/inceptionv3_fp32_pretrained_model.pb

WORKDIR /tf
CMD ["/bin/bash"]
```

- 运行以下命令构建 Docker 映像，并将其推送到 Docker 注册表（请根据您的环境调整值）：

```
docker build -f dlrs.Dockerfile -t stacks-dlrs-mkl:v0.5.0 .
docker login -u "${DOCKER_REPO_USER}" -p "${DOCKER_REPO_PASS}" "${DOCKER_REPO}"
docker tag stacks-dlrs-mkl:v0.5.0 ${DOCKER_REPO_PROJECT}/stacks-dlrs-mkl:v0.5.0
docker push ${DOCKER_REPO_PROJECT}/stacks-dlrs-mkl:v0.5.0
```

- Helm 3 就绪后，运行 `helm create dlrs-benchmark` 命令创建一个 Helm 3 图表，以便使用深度学习参考堆栈容器部署 TensorFlow 基准测试。

5. 在 dlr-benchmark 目录中，编辑以下文件，复制/黏贴以下内容：

values.yaml:

```
image:
  repository: ${DOCKER_REPO_PROJECT}/stacks-dlr-mkl
  tag: v0.5.0

# How many jobs run in parallel on K8s
jobs: 5

# How many resources apply to pod in Guaranteed QoS class (requests==limits)
resources:
  cpu: «16»
  memory: «8Gi»
```

Chart.yaml:

```
apiVersion: v2
name: dlr-benchmark
description: A Helm chart for Kubernetes
type: application
version: 0.1.0
appVersion: 1.0.0
```

templates/job.yaml:

```
apiVersion: batch/v1
kind: Job
metadata:
  labels:
    role: dlr-benchmark
    name: dlr-benchmark
spec:
  completions: {{ .Values.jobs }}
  parallelism: {{ .Values.jobs }}
  template:
    metadata:
      labels:
        role: dlr-benchmark
        name: dlr-benchmark
    spec:
      containers:
        - name: dlr-benchmark
          image: {{ .Values.image.repository }}:{{ .Values.image.tag }}
          imagePullPolicy: Always
          resources:
            requests:
              cpu: {{ .Values.resources.cpu }}
              memory: {{ .Values.resources.memory }}
            limits:
              cpu: {{ .Values.resources.cpu }}
              memory: {{ .Values.resources.memory }}
          command:
            - «bash»
          args:
            - «/usr/local/bin/job.sh»
          volumeMounts:
            - name: usr-local-bin
              mountPath: «/usr/local/bin/job.sh»
              subPath: job.sh
          restartPolicy: Never
      volumes:
        - name: usr-local-bin
          configMap:
            name: dlr-benchmark-job
```


templates/configmap.yaml:

```

apiVersion: v1
kind: ConfigMap
metadata:
  name: dlrs-benchmark-job
data:
  job.sh: |
    #!/bin/bash

    #####
    # Run benchmark.
    # Arguments:
    #   $1 Filepath to model
    #   $2 Model name
    #   $3 Precision of the model
    #####
    function run_benchmark() {
    for batch_size in 1 16 32 64 128
    do
        benchmark_app_output=$(python3 /tf/intel-models/benchmarks/launch_benchmark.py --in-graph $1 --model-
name $2 --framework tensorflow --precision $3 --mode inference --batch-size ${batch_size} --benchmark-only 2>&1)
        latency=$(echo $benchmark_app_output | sed -n <s/. *Latency: \([0-9]*\.[0-9]*\) ms.*\/1/p>)>
        if [[ -z «${latency}» ]]; then
            latency=>-1<
        fi
        fps=$(echo $benchmark_app_output | sed -n <s/. *[T,t]hroughput.*: \([0-9]*\.[0-9]*\) images\/sec.*\/1/p>)>
        if [[ -z «${fps}» ]]; then
            fps=>-1<
        fi
        echo «$2_$3,${batch_size},${fps},${latency}»
    done
    }

    echo «model_name,batch_size,fps,latency_ms»

    run_benchmark «/tf/resnet50v1_5_int8_pretrained_model.pb» «resnet50v1_5» «int8»
    run_benchmark «/tf/resnet50_v1.pb» «resnet50v1_5» «fp32»

    run_benchmark «/tf/inceptionv3_int8_pretrained_model.pb» «inceptionv3» «int8»
    run_benchmark «/tf/inceptionv3_fp32_pretrained_model.pb» «inceptionv3» «fp32»

```

6. 准备好 dlrs-benchmark Helm 图表后，在 Kubernetes 集群中通过以下命令运行该图表：

```
helm install --namespace default --name tf-benchmark tf-benchmark
```

7. 该命令完成后，集群中将生成 Kubernetes 作业。运行以下命令获取基准测试结果：

```
kubect1 get pods -l job-name=dlrs-benchmark -o name | xargs -n 1 kubect1 logs
```

VMware 环境中的 DataRobot 配置

要在您的环境中安装 DataRobot，请[联系 DataRobot 支持人员](#)进行咨询。他们会提供所有必需的文件和相关说明，协助您进行符合您需求的部署。表 A1 列出了本例中使用的虚拟机及规格。

表 A1. 参考架构 DataRobot 规格

虚拟机名称	数量	vCPU 内核数量	RAM	存储
应用和数据	1	8	64 GB	500 GB
建模服务	2	48	384 GB	500 GB
预测服务器	1	4	32 GB	500 GB
模型管理	1	4	16 GB	500 GB
人工智能目录	1	4	32 GB	500 GB

Kubeapps 的使用

要使用 Kubeapps，您必须根据[安装文档](#)所述在您的环境中安装 Helm 3。安装完成后，请执行以下步骤：

1. 按照此[文档](#)所述在您的 Kubernetes 集群中安装 Kubeapps。
2. 运行 `helm create [NAME_OF_YOUR_HELM_CHART]` 命令创建 Helm 图表。准备图表所需的所有文件。有关如何创建 Helm 图表的更多信息，请参阅 Helm 官方[文档](#)。
3. 准备好 Helm 图表后，运行 `helm package [CHART_PATH]` 命令打包图表。这样会创建一个 .tgz 文件。
4. 登录 Harbor 注册表，然后把您的 Helm 图表上传到目标项目。有关在 Harbor 中管理 Helm 图表的更多信息，请参阅本[文档](#)。
5. 现在，您已经可以在 Kubeapps 中添加 Harbor Helm 存储库了：
 - a. 如要添加您自己的存储库，您可返回 Kubeapps 用户界面，选择“**Configuration → App Repositories**”，然后单击“**Add App Repository**”。
 - b. 在相应的字段中提供 Harbor 注册表地址，然后接受更改。
 - c. 创建存储库后，单击您想使用的存储库的链接，然后使用 Kubeapps 部署自己的应用。

附录 B: Windows 系统上 Microsoft SQL 基准测试配置

基础配置中的 Microsoft SQL 虚拟机

vCPU: 8
RAM: 44 GB
网卡: VMXNET 3
虚拟机: 每个 ESXi 节点最多五台虚拟机, 总共 20 台
虚拟机存储: vSAN

增强配置中的 Microsoft SQL 虚拟机

vCPU: 8
RAM: 44 GB
网卡: VMXNET 3
虚拟机: 每个 ESXi 节点最多八台虚拟机, 总共 32 台
虚拟机存储: vSAN
数据库存储: 英特尔® 傲腾™ 持久内存

磁盘布局

图 B1 是 Microsoft SQL 的磁盘布局:

- 4 个磁盘用于存储数据文件 (每个磁盘 18 GB)
- 2 个磁盘用于存储 TempDB 文件 (每个磁盘 200 MB)
- 1 个磁盘用于存储 TempLog (大小为 100 MB)
- 1 个磁盘用于存储事务日志 (大小为 10 GB) 。

英特尔® 傲腾™ 持久内存上的数据磁盘应格式化为 NTFS 并配置为 DAX (直接访问) 模式。英特尔® 傲腾™ 持久内存上的日志磁盘应格式化为 NTFS 并配置为块访问模式。











	OS (C:)	本地磁盘	99.3 GB	20.4 GB
	data1 (F:)	本地磁盘	17.9 GB	64.1 MB
	data2 (G:)	本地磁盘	17.9 GB	128 MB
	data3 (H:)	本地磁盘	17.9 GB	128 MB
	data4 (I:)	本地磁盘	17.9 GB	128 MB
	tempdb_1 (J:)	本地磁盘	182 MB	138 MB
	tempdb_2 (K:)	本地磁盘	182 MB	138 MB
	temp_log (L:)	本地磁盘	182 MB	162 MB
	tran_log (M:)	本地磁盘	9.97 GB	1.60 MB
	mssqlbackup (R:)	本地磁盘	499 GB	201 GB

图 B1. Microsoft SQL 虚拟机测试机的磁盘命名、布局 and 大小示例

Microsoft SQL 数据库配置

数据库数据拆分为 16 个文件，分布在四个磁盘上（每个磁盘四个文件）。

T-SQL (Transact-SQL) 的其他选项:

```
ALTER DATABASE [benchmarkedDB] SET DISABLE_BROKER WITH ROLLBACK
IMMEDIATE
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [benchmarkedDB].[dbo].[sp_fulltext_database] @action =
'enable'
end
GO
ALTER DATABASE [benchmarkedDB] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [benchmarkedDB] SET ANSI_NULLS OFF
GO
ALTER DATABASE [benchmarkedDB] SET ANSI_PADDING OFF
GO
ALTER DATABASE [benchmarkedDB] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [benchmarkedDB] SET ARITHABORT OFF
GO
ALTER DATABASE [benchmarkedDB] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [benchmarkedDB] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [benchmarkedDB] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [benchmarkedDB] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [benchmarkedDB] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [benchmarkedDB] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [benchmarkedDB] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [benchmarkedDB] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [benchmarkedDB] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [benchmarkedDB] SET AUTO_UPDATE_STATISTICS_ASYNC
OFF
GO
ALTER DATABASE [benchmarkedDB] SET DATE_CORRELATION_OPTIMIZATION
OFF
GO
ALTER DATABASE [benchmarkedDB] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [benchmarkedDB] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [benchmarkedDB] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [benchmarkedDB] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [benchmarkedDB] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [benchmarkedDB] SET RECOVERY FULL
GO
ALTER DATABASE [benchmarkedDB] SET MULTI_USER
GO
ALTER DATABASE [benchmarkedDB] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [benchmarkedDB] SET DB_CHAINING OFF
GO
ALTER DATABASE [benchmarkedDB] SET FILESTREAM( NON_TRANSACTED_
ACCESS = OFF )
GO
```

```
ALTER DATABASE [benchmarkedDB] SET TARGET_RECOVERY_TIME = 60
SECONDS
GO
ALTER DATABASE [benchmarkedDB] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [benchmarkedDB] SET QUERY_STORE = OFF
GO
USE [benchmarkedDB]
GO
ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 0;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET MAXDOP =
PRIMARY;
GO
ALTER DATABASE SCOPED CONFIGURATION SET LEGACY_CARDINALITY_
ESTIMATION = OFF;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET LEGACY_
CARDINALITY_ESTIMATION = PRIMARY;
GO
ALTER DATABASE SCOPED CONFIGURATION SET PARAMETER_SNIFFING = ON;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET PARAMETER_
SNIFFING = PRIMARY;
GO
ALTER DATABASE SCOPED CONFIGURATION SET QUERY_OPTIMIZER_HOTFIXES
= OFF;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET QUERY_
OPTIMIZER_HOTFIXES = PRIMARY;
GO
ALTER DATABASE [benchmarkedDB] SET READ_WRITE
GO
USE [benchmarkedDB]
GO
IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is_default=1
AND name = N'SECONDARY') ALTER DATABASE [benchmarkedDB] MODIFY
FILEGROUP [SECONDARY] DEFAULT
GO
USE [benchmarkedDB]
GO
declare @autogrow bit
SELECT @autogrow=convert(bit, is_autogrow_all_files) FROM sys.
filegroups WHERE name=N'SECONDARY'
if(@autogrow=0)
ALTER DATABASE [benchmarkedDB] MODIFY FILEGROUP [SECONDARY]
AUTOGROW_ALL_FILES
GO
USE [master]
GO
declare @autogrow bit
SELECT @autogrow=convert(bit, is_autogrow_all_files) FROM sys.
filegroups WHERE name=N'SECONDARY'
if(@autogrow=0)
ALTER DATABASE [benchmarkedDB] MODIFY FILEGROUP [SECONDARY]
AUTOGROW_ALL_FILES
GO
-- disable autogrow on tranlog
ALTER DATABASE [benchmarkedDB]
MODIFY FILE (NAME = N'benchmarkedDB_log', FILENAME = 'M:\vdap_
log.ldf', SIZE = 10000MB, FILEGROWTH = 0 );
GO
```


Microsoft SQL 配置

作为 TSQL:

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE WITH OVERRIDE;
GO
sp_configure 'max server memory', 50000;
GO
RECONFIGURE WITH OVERRIDE;
GO
sp_configure 'min server memory', 40000;
GO
RECONFIGURE WITH OVERRIDE;
GO
sp_configure 'max worker threads', 3000;
GO
RECONFIGURE WITH OVERRIDE;
GO
sp_configure 'recovery interval', 32767;
GO
RECONFIGURE WITH OVERRIDE;
GO
sp_configure 'lightweight pooling', 1;
GO
RECONFIGURE WITH OVERRIDE;
GO
sp_configure 'priority boost', 1;
GO
RECONFIGURE WITH OVERRIDE;
GO
EXEC sys.sp_configure N'network packet size (B)', N'8192'
GO
RECONFIGURE WITH OVERRIDE
GO
ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED HYBRID_BUFFER_POOL = ON;
GO
ALTER DATABASE [benchmarkedDB] SET MEMORY_OPTIMIZED = ON;
GO
```

用于负载生成的 HammerDB 配置

- Windows SQL Server 2019 15.0.2070.41, 补丁更新至 2020-03-01
- Windows Server 2019 Datacenter 17763.rs5_release.180914-1434, 补丁更新至 2020-03-01
- 网卡: VMXNET3
- 存储: vSAN
- 网络: 基于 VLAN (不含 NSX-T 覆盖网络分段)
- 用于基准测试的软件: HammerDB 3.3 Windows
- SQL Server ODBC 驱动程序: 适用于 SQL Server 的 ODBC Driver 17
- TPC-C 驱动程序脚本: 时控驱动程序脚本
- 每用户总事务数: 1,000,000
- 启动时长 (分钟) : 10
- 测试时长 (分钟) : 20
- 使用所有仓库: 是
- 每个虚拟机 750 个仓库和 75 个用户
- 每个 Microsoft SQL 虚拟机 1 个 HammerDB 虚拟机
- 采集的数据: 所有 HammerDB 的每分钟事务处理量 (TPM) 总和 (所有 Microsoft SQL 实例) ; 所有 Microsoft SQL 实例的“CPU 时间: 请求 (5 ms)”和“CPU 时间: 总时间 (20 ms)”

尾注

- ¹ 英特尔于 2020 年 6 月 15 日至 23 日期间进行的测试。每个集群（基础集群、增强集群和管理集群，用于生成工作负载）由四台机器组成。给定集群中的所有机器均相同。虚拟机部署在这些集群上：基础集群的每个节点部署五台虚拟机，增强集群的每个节点部署八台虚拟机。我们分别为基础配置和增强配置环境设置了四个场景。第一个场景仅使用集群中的一个节点，第二个场景使用两个节点，依此类推，直到集群的全部四个节点都投入使用。每个场景重复测试三次；本文给出的结果为每个场景三次测试的平均值。
- 4 节点基础集群配置：**英特尔® 至强® 6248 处理器（2.5 GHz，20 核）；384 GB DRAM（12 x 32 GB 2,933 MHz）；启用英特尔® 超线程技术；启用英特尔® 睿频加速技术；存储：VMware vSAN，磁盘组：1x 英特尔® 傲腾™ 固态硬盘 DC P4800X（375 GB）和 2x 英特尔® 固态硬盘 DC P4510（2 TB），每节点 2 个磁盘组；BIOS = 02.01.0010；微代码 = 0x0500002c；操作系统 = VMware ESXi 6.7.0 15160138；1x 英特尔® 以太网适配器 XXV710-DA2。
- 基础集群中的虚拟机配置：**8 个 vCPU；44 GB vMemory；vDisks: 100 GB - 操作系统驱动器（LSI 控制器），4 x 18 GB - Data DB（1 个 SCSI 控制器），每个磁盘 4 个文件，2 x 200 MB - TempDB（1 个 SCSI 控制器），200 MB - TempLog（与用户事务日志共享 SCSI 控制器），10 GB - 用户事务日志（1 个 SCSI 控制器）；vNetwork = VMXNET3；操作系统 = Windows Server 2019 Datacenter 17763.rs5.180914-1434；数据库 = MSSQL Server 2019 15.0.2070.41。
- 4 节点增强集群配置：**英特尔® 至强® 8268 处理器（2.9 GHz，24 核）；384 GB DRAM（12 x 32 GB 2,933 MHz）和 1,536 GB 英特尔® 傲腾™ 持久内存（12 x 128 GB）；启用英特尔® 超线程技术；启用英特尔® 睿频加速技术；存储：VMware vSAN，磁盘组：1x 英特尔® 傲腾™ 固态硬盘 DC P4800X（375 GB）和 2x 英特尔® 固态硬盘 DC P4510（2 TB），每节点 2 个磁盘组；BIOS = 02.01.0010；微代码 = 0x0500002c；操作系统 = VMware ESXi 6.7.0 15160138；1x 英特尔® 以太网适配器 XXV710-DA2；持久内存使用 2-2-2 App Direct 模式配置；持久内存固件版本 = 01.02.00.5417。
- 增强集群中的虚拟机配置：**2 个 vCPU；44 GB vMemory；vDisks: 100 GB - 操作系统驱动器（LSI 控制器），4 x 18 GB Data DB（NVDIMM），每个磁盘 4 个文件，2 x 200 MB TempDB（NVDIMM 非 DAX 卷），200 MB TempLog（NVDIMM 非 DAX 卷）；10 GB 用户事务日志（NVDIMM 非 DAX 卷）；vNetwork = VMXNET3；操作系统 = Windows Server 2019 Datacenter 17763.rs5.180914-1434；数据库 = MSSQL Server 2019 15.0.2070.41。
- ² 该独立测试/基准测试的结果已发布于 glesys.se/blogg/benchmarking-intel-optane-dc-persistent-memory（表 3，以 512 字节的块大小测得时延达到微秒级）。
- ³ 英特尔。“Product Brief: Intel Optane SSD DC P4800X/P4801X Series”（产品简介：英特尔® 傲腾™ 固态硬盘 DC P4800X/P4801X 系列）：intel.com/content/www/cn/zh/products/docs/memory-storage/solid-state-drives/data-center-ssds/optane-ssd-dc-p4800x-p4801x-brief.html。
- ⁴ 英特尔。“Product Brief: Intel Optane SSD DC P4800X Series”（产品简介：英特尔® 傲腾™ 固态硬盘 DC P4800X 系列）：intel.com/content/www/cn/zh/solid-state-drives/optane-ssd-dc-p4800x-brief.html。基于英特尔内部测试。
- ⁵ 英特尔® 以太网 700 系列包括经过广泛测试的网络适配器、配件（光学元件和电缆）、硬件、软件以及广泛操作系统支持。有关产品组合解决方案的完整列表，请访问：intel.cn/ethernet。硬件和软件已在英特尔® 至强® 可扩展处理器和网络生态系统中经过全面验证。产品针对英特尔® 架构和广泛的操作系统生态进行了优化，包括 Windows、Linux 内核、FreeBSD、Red Hat Enterprise Linux (RHEL)、SUSE、Ubuntu、Oracle Solaris 以及 VMware ESXi。
- ⁶ 英特尔® 以太网 700 系列网络适配器的客户售前和售后服务由全球支持体系提供支持。
- ⁷ 英特尔® 以太网 700 系列网络适配器支持的连接和介质类型包括直连铜线、光纤 SR/LR（QSFP+、SFP+、SFP28、XLPP1/CR4、25G-CA/25G-SR/25G-LR）、双绞铜线（1000BASE-T/10GBASE-T）和底板（XLAUI/XAUI/SFI/KR/KR4/KX/SGMII）。请注意，英特尔可提供 QSFP+ 介质类型。
- ⁸ 英特尔® 以太网 700 系列网络适配器支持的速度包括 1 GbE、10 GbE、25 GbE 和 40 GbE。
- ⁹ docs.vmware.com/cn/VMware-Cloud-on-AWS/services/com.vmware.vmc-aws.getting-started/GUID-EF198D55-03E3-44D1-AC48-6E2ABA31FF02.html。英特尔部署了四次 VMware Cloud on AWS，根据以往经验，可以确定每次部署大约需要两个小时。
- ¹⁰ software.intel.com/content/www/cn/zh/develop/tools/distribution-for-python/benchmarks.html。
- ¹¹ 有关更多信息，请访问 builders.intel.com/ai/blog/1-billion-models-datarobot-intel。
- ¹² 参见尾注 1。
- ¹³ 英特尔于 2020 年 6 月 15 日至 23 日期间进行的测试。有关物理基础设施配置信息，请参见尾注 1。
- ¹⁴ 英特尔于 2020 年 6 月 15 日至 23 日期间进行的测试。有关物理基础设施配置信息，请参见尾注 1。
- ¹⁵ 英特尔。“Technology Brief: Intel Optane Technology: Memory or Storage? Both”（技术简介：英特尔® 傲腾™ 技术：内存还是存储？两者兼备）：intel.com/content/dam/www/public/us/en/documents/technology-briefs/what-is-optane-technology-brief.pdf。
- ¹⁶ 英特尔于 2020 年 6 月 15 日至 23 日期间进行的测试。工作负载：fp32 精度 ResNet 50 v1.5 拓扑。默认 TensorFlow: tensorflow/tensorflow:1.15.0-py3；优化 TensorFlow: clearlinux/stacks-dlrs-mkl:v0.5.0。
- 4 节点基础集群配置：**英特尔® 至强® 6248 处理器（2.5 GHz，20 核）；384 GB DRAM（12 x 32 GB 2,933 MHz）；启用英特尔® 超线程技术；启用英特尔® 睿频加速技术；存储：VMware vSAN，磁盘组：1x 英特尔® 傲腾™ 固态硬盘 DC P4800X（375 GB）和 2x 英特尔® 固态硬盘 DC P4510（2 TB），每节点 2 个磁盘组；BIOS = 02.01.0010；微代码 = 0x0500002c；操作系统 = VMware ESXi 6.7.0 15160138；1x 英特尔® 以太网适配器 XXV710-DA2。
- 基础集群中的胖虚拟机配置：**80 个 vCPU；操作系统 = CentOS Linux 版本 8.1.1911；内核 = 4.18.0-147.5.1.el8_1.x86_64；其他软件：VMware vSphere 6.7、VMware Cloud Foundation 3.9 和 VMware ESXi 虚拟机管理程序 6.7 更新 3。
- 基础集群归一化结果：**相较于默认 TensorFlow 性能，使用深度学习参考堆栈版本的 TensorFlow，吞吐量提升 2.2 倍以上。
- 4 节点增强集群配置：**英特尔® 至强® 8268 处理器（2.9 GHz，24 核）；384 GB DRAM（12 x 32 GB 2,933 MHz）和 1,536 GB 英特尔® 傲腾™ 持久内存（12 x 128 GB）；启用英特尔® 超线程技术；启用英特尔® 睿频加速技术；存储：VMware vSAN，磁盘组：1x 英特尔® 傲腾™ 固态硬盘 DC P4800X（375 GB）和 2x 英特尔® 固态硬盘 DC P4510（2 TB），每节点 2 个磁盘组；BIOS = 02.01.0010；微代码 = 0x0500002c；操作系统 = VMware ESXi 6.7.0 15160138；1x 英特尔® 以太网适配器 XXV710-DA2；持久内存使用 2-2-2 App Direct 模式配置；持久内存固件版本 = 01.02.00.5417。
- 增强集群中的胖虚拟机配置：**96 个 vCPU；或者与基础集群中的胖虚拟机相同。
- 增强集群归一化结果：**相较于默认 TensorFlow 性能，使用深度学习参考堆栈版本的 TensorFlow，吞吐量提升 2.5 倍以上。

¹⁷ 英特尔于 2020 年 6 月 15 日至 23 日期间进行的测试。工作负载：fp32 精度 Inception V3 拓扑。有关物理基础设施配置和胖虚拟机配置信息，请参见尾注 16。

基础集群归一化结果：相较于默认 TensorFlow 性能，使用深度学习参考堆栈版本的 TensorFlow，吞吐量提升 2.4 倍以上。

增强集群归一化结果：相较于默认 TensorFlow 性能，使用深度学习参考堆栈版本的 TensorFlow，吞吐量提升 3 倍以上。

¹⁸ 英特尔于 2020 年 6 月 15 日至 23 日期间进行的测试。工作负载：fp32 精度 Inception V3 拓扑。6 台工作虚拟机，每台 16 个 vCPU。有关物理基础设施配置，请参见尾注 16。默认 TensorFlow：tensorflow/tensorflow:1.15.0-py3；优化 TensorFlow：clearlinux/stacks-dlrs-mkl:v0.5.0。

工作节点配置：操作系统 = Ubuntu 16.04.6 LTS；内核 = 4.15.0-55-generic；其他软件：VMware vSphere 6.7、VMware Cloud Foundation 3.9 和 VMware ESXi 虚拟机管理程序 6.7 更新 3。

归一化结果：

工作虚拟机数量	吞吐量提升
1	1 倍
2	1.98 倍
3	2.33 倍
4	3.03 倍
5	3.42 倍
6	4.17 倍

¹⁹ 有关详细信息，请访问 <https://pypi.org/project/datarobot>。

²⁰ 有关详细信息，请访问 <https://cran.r-project.org/web/packages/datarobot/index.html>。

解决方案来自：



英特尔技术特性和优势取决于系统配置，并可能需要启用硬件、软件或激活服务。更多信息请从原始设备制造商或零售商处获得，或请见 intel.cn。没有任何产品或组件是绝对安全的。

性能测试中使用的软件和工作负荷可能仅在英特尔微处理器上进行了性能优化。诸如 SYSmark 和 MobileMark 等测试均系基于特定计算机系统、硬件、软件、操作系统及功能。上述任何要素的变动都有可能导致测试结果的变化。请参考其他信息及性能测试（包括结合其他产品使用时的运行性能）以对目标产品进行全面评估。更多信息，详见 www.intel.cn/benchmarks。

英特尔编译器针对英特尔微处理器的优化程度可能与针对非英特尔微处理器的优化程度不同。这些优化包括 SSE2、SSE3 和 SSSE3 指令集和其他优化。对于非英特尔微处理器上的任何优化是否存在、其功能或效力，英特尔不做任何保证。本产品中取决于微处理器的优化是针对英特尔微处理器。不具体针对英特尔微架构的特定优化为英特尔微处理器保留。

描述的成本降低情景均旨在特定情况和配置中举例说明特定英特尔产品如何影响未来成本并提供成本节约。情况均不同。英特尔不保证任何成本或成本降低。

英特尔并不控制或审计第三方数据。请您审查该内容，咨询其他来源，并确认提及数据是否准确。

英特尔是 BenchmarkXPRT 开发社区的赞助商和成员，并且曾是 XPRT 基准测试系列的主要开发商。Principled Technologies 是 XPRT 基准测试系列的出版商。请参阅其他信息和性能测试以对预期购买产品进行全面评估。

英特尔、英特尔标识以及其他英特尔商标是英特尔公司在美国和其他国家的商标。

其他的名称和品牌可能是其他所有者的资产。© 英特尔公司版权所有 0820/SALL/KC/PDF