白皮书

推动企业数据中心向现代化环境转型：以 Kubernetes 作为混合云基础

赞助商：IBM

Gary Chen　　Al Gillen

2019 年 11 月

IDC 观点

现今的 IT 从业人员身处变换不断的 IT 世界。核心技术几年间即会更替，并伴随着广泛而深入的典型企业计算环境变革。最近，基于云的部署模型概念在 IT 行业得到广泛认可，被认为是理想的长期部署方案。但是，传统本地计算环境包含服务器架构、操作系统 (OS) 、虚拟化以及平台服务，要想将应用和数据直接从中迁移至以云为中心的部署环境并非易事，面临重重挑战。

对于诸多企业而言，采用混合云计算模型，为客户打造更为顺畅的本地资源和公有云资源访问体验，是从本地计算模型向以公有云为中心的平台转型最可行的途径。通过这种方式，企业可填补传统计算架构与现代计算架构间的鸿沟，规避步子太大的风险，循序渐进，稳步推进关键工作负载向公有云完全迁移。可喜的是，IT 行业早已踏上技术探索之旅，为客户赋能，助力他们搭建涵盖本地资源与远程资源常用运营要素的混合云。

打造可移植的多云平台是实现混合云的关键一步。IT 行业以 Linux、容器以及 Kubernetes 为通用抽象化层的基础，向该方向进发。Amazon Web Services (AWS)、Google、IBM Cloud、Microsoft、Pivotal、Red Hat 以及 VMware 与其他众多供应商携手开发此类平台，让客户使用服务器和云混合服务以及混合位置成为可能。

近日，IBM 发布了下一代 LinuxONE III 系统，旨在强化混合云环境的私有云部分。该平台性能更为强大，在芯片上配备了全新的压缩加速器，可有效提升普遍加密能力,实现广泛的软件集成。通过使用 LinuxONE 系统，Red Hat OpenShift 可以成为广泛可用的软件资源；如果与 IBM 一系列全新的开发与部署服务 Cloud Paks 结合使用，则有助于加快现代开发工具、中间件、人工智能 (AI) 数据服务以及管理资源的部署速度。

市场概况

尽管业界努力让信息技术化繁为简，但每一次简化尝试都会导致繁上加繁，因为简化在本质上是递增的。这一定律同样适用于云计算的转变历程。现在，IDC 将云计算本身视为紧密相关的技术集合，包含如下方面：

- **本地私有云：**本地私有云融合传统软硬件，可用作传统企业服务器。不仅如此，从管理平面视角来看，本地私有云还可提供近似公有云环境的使用体验。在理想情况下，本地私有云系统将拥有近似公有云资源的运行方式，也就是说，客户无需刻意学习，即可迁移至混合云部署方案。
- **远程公有云：**远程公有云是由超大型提供商或小型云服务供应商提供的第三方云环境。例如，阿里云、AWS、Google、IBM Cloud 以及 Microsoft Azure。此外，公有云服务由纯软件解决方案提供，部署于多个超大型云提供商环境中，例如 Pivotal Cloud Foundry（以及 Cloud Foundry Foundation 开源软件解决方案）、Red Hat OpenShift、SUSE Cloud 以及 VMware Cloud。
- **混合云：**混合云更像是完全不同的技术用例。混合云服务可为企业提供跨多个云运营的能力，包括本地私有云和公有云，或使用多个公有云（通常称为多云），以此作为公共控制平台管理的公用基础架构。大多公有云供应商都在致力于制定混合云战略。这些战略本质上通常不具备平台多样性。与之相反，某些纯软件解决方案（包括 Pivotal、Red Hat、VMware 以及 SUSE 提供的方案）则同时支持混合云和多平台混合云部署方案。

然而，混合环境中的本地部分可能并不具备同等的向外扩展属性，同时，对开发者具有吸引力的某些公有云服务可能无法在本地提供（即使仍可通过远程使用，假设延迟没问题）。在这种情况下，具备私有云的混合部署方案便成为了更具吸引力也更加可行的选择。对于许多客户而言，迁移至包含强大私有云组件的混合云更合其意。

长久以来，客户一直都在担忧应用的可移植性。然而，IT 行业在攻克这道难题方面取得了重大进展，解决之道如下：

- Python 以及 JavaScript 等解释语言使用量增长。以前，由于编译器仅对特定平台最大限度优化了性能，编译语言广为流行。如今，计算资源充足，性能不再是制约解释语言使用的因素。现代持续集成/持续交付 (CI/CD) 管道经过配置，对于使用中的编译语言，可为不同部署方案创建多组二进制文件。
- 容器目前是一种在不同基础架构、云和系统架构中广泛使用的抽象层。无论是新应用还是现有应用，不论是否设计为微服务集，均使用容器重新打包。如今，容器还具备多架构支持功能。

- IT 行业正向 Kubernetes 作为下一代应用的底层容器管理和统筹层发展。目前，已有多个同时面向公有云和私有云部署的可移植 Kubernetes 环境面世。随着关键系统和云端的持续性 Kubernetes 平台的面世，构建混合云环境来涵盖不同物理位置的资源成为了现实。

并非所有容器包含的软件都可被描述为新型微服务。实际上，容器中托管的应用多种多样，相应也需要多种多样的外部服务。举例而言，有些容器会包含已重新托管的整体应用，而另一些容器则可能会部分重构有状态应用，安全性要求更高，且具备反映其来源环境的其他属性。这些应用需要可完全满足其需求的系统和基础架构软件。

容器和 Kubernetes

容器化 Linux 应用的概念由来已久，它最初是在 Linux 容器 (LXC) 技术下实现的，是 Linux 操作系统的组成部分。然而，LXC 的采用十分受限，直到 Docker（既指公司也指技术）面世，创造出强大易用的容器打包应用技术，这一限制才迎刃而解。

Docker 和 Open Container Initiative (OCI) 容器将应用及其所有依赖项统统打包到单个可移植容器映像中。然后，这些映像在集中式容器注册表中共享。在该注册表中，其他开发者可对它们进行迭代或将其推送至生产环境。容器映像也会在映像中使用层概念，便于开发者在现有映像之上轻松构建。执行容器时，这些容器会在各自的沙箱中运行，这样一来，每个容器都相互隔离，似乎也都具有完整的操作系统。开发者友好型工具与 API 可让容器构建、共享及运行变得轻松高效，正因如此，此类工具与 API 不久便风靡业界。

容器化应用仅仅是第一步，复杂容器化服务组合运营之道尚需探索。Google 借助自身内部 Borg 容器统筹技术概念及技术积累，在 IBM 、Red Hat 以及业界其他公司的积极参与协助下，打造出开源软件 Kubernetes 项目。Google 将 Kubernetes 项目移交给新成立的 Cloud Native Compute Foundation (CNCF) 组织监管，确保该项目完全开源，为 IT 行业共享。随后，IT 行业将 Kubernetes 认定为首选下一代基础架构以及容器化应用的部署与统筹平台。

作为可移植的通用应用打包标准，容器日益风靡业界。探其成功缘由，其一在于容器得以标准化，成为 Open Containers Initiative (OCI) 的组成部分。OCI 对容器执行方式（运行时）以及容器映像格式化方式进行了定义。标准化也意味着在各种容器平台和云服务之间，容器具备一致性、互操作性和可移植性。此外，标准化还进一步扩展了 Kubernetes 堆栈，实现了统筹与管理。

虽然 Kubernetes 并非 OCI 那样的法规性标准，但它是一个独立监管的开源项目，业界参与十分广泛。因此，容器不仅具备标准化格式，还拥有大量在实施中保持一致的管理栈。诸如 Istio 服务网格以及适用于无服务器的 Knative 之类的其他组件也开始受到关注，有朝一日也会得到广泛应用，为业界提供通用度更高的堆栈。虽然标准化确实加速了容器的采用，但容器在业界掀起波澜的首要原因还是在于，无论是在开发生命周期还是 DevOps 支持的部署中，容器都助力开发者实现了大幅提速。

对于开发者而言，容器无疑是理想之选，既可以高效封装全新的云原生微服务，也可以通过使用 CI/CD，将这些变更下推至日渐自动化的软件构建管道中。开发者友好型 API 提升了复杂软件的操作效率和便捷程度，优化了开发者工作流。此外，得益于包含了所有依赖项，容器可助力业界打造更多自动化测试系统，加大环境控制力度，从而改善代码质量。最终，容器提升了软件开发以及变更部署的速度，提高了开发者的生产力。

对于运营者而言，容器和 Kubernetes 为他们提供了高度现代化、扩展灵活且自动化的大型网站级应用运行方式。Kubernetes 融入了大型网络公司在大规模稳定运行快速更替的应用方面积累的知识与经验。它建立了多种部署模式，如蓝/绿升级、适用于新应用功能的 A/B 测试，以及多项自动化扩展选项。此外，容器样式部署得益其不可变基础架构，可助力解决配置难题。这意味着容器状态一经在其映像中定义，在运行时期间就绝对不会发生变化。若要进行任何变更，就要关闭旧的容器实例，然后启动一个新映像，而不是对运行中的映像安装补丁或更改配置。不仅如此，容器存储库还有助于集中管理容器映像，维护版本。容器的轻量化以及反应式特性，加上现代控制平台，可支持 IT 高效部署及管理现代化应用。

随着客户转投混合云和多云环境，容器定会在跨不同环境的可移植性和一致性方面发挥关键作用。如前文所言，随着 IT 行业对 OCI 以及 Kubernetes 控制平台的广泛采用，容器平台核心上会不断趋近。CNCF 提供 Kubernetes 符合性测试与认证服务，换而言之，所有 Kubernetes 均需在核心功能层面保持一致。这意味着客户可以大量使用任何 Kubernetes 产品，也有望实现一定程度的兼容性。

不论发行版本、云服务或底层基础架构如何，开发者均可使用合意的 API 和工具自如处理容器与 Kubernetes。这有助于跨本地云及各类公有云提供始终如一的开发者环境。此外，在容器界面保持不变的情况下，不论底层的硬件如何，容器都有助于跨不同系统架构实现抽象化，开发者无需深入了解系统或操作系统的细节，便可为之开发应用。对于管理 Kubernetes 的运营者而言，他们需要整合一些系统专用知识，安装一些部署工具，但是在任何发行版本间，Kubernetes 的操作方式及其管理的应用均在很大程度上保持一致。

多架构容器与混合云

通过使用容器，开发者无需关注各种底层系统之间的差异。然而，像 IBM LinuxONE 这样的企业平台可以在容器的运维与应用方面带来额外价值，容器开发者无需学习太多新技能便能轻松使用容器。

IBM LinuxONE 包含基于固件的虚拟机管理器，它与 z/VM 或 KVM 软件虚拟机管理器协同工作，安全灵活地配置计算、内存以及 I/O 资源。如今，大多数容器都运行于虚拟机 (VM) ，而非裸机统筹环境。虽然这些技术似乎有所重叠，但它们大多数都在截然不同的层级上运行。虚拟机管理器负责硬件虚拟化及分区，而容器则负责操作系统虚拟化。

在虚拟机中运行容器具备多项优势：

- 由于虚拟机边界比容器边界更为清晰，因此在虚拟机中运行容器可提供额外的隔离层。
- 将当前大型硬件系统分割成更多的可消费区块（此举无须将大量容器整合至单一操作系统内核，亦可提升安全性和可靠性）。
- 支持更灵活的不同操作系统组合，甚至是同一操作系统的不同版本/补丁级别的组合。

IBM LinuxONE 平台中基于固件的独特虚拟机管理器可帮助 Kubernetes 实现灵活扩展。借助虚拟机管理器，IBM LinuxONE 能够在不中断的情况下实现向上扩展和向外扩展。举例而言，这项功能作为 Kubernetes 垂直 Pod 自动扩展功能的补充，可帮助优化容器的 CPU 和内存配置情况。

随着企业向微服务架构转移，容器化的内容大多是可能会也可能不会重构的传统应用。将现有应用容器化仍具备价值，不少企业已经开始广泛开展这项工作，其中约半数的容器化工作涉及传统应用。IBM LinuxONE 及其虚拟机管理器既灵活又可靠，因此适合在这个迁移过程中托管大型单体式容器。

最新版本的 IBM LinuxONE 系统为开发者和管理员提供了易于使用的全新软件解决方案，称为 IBM Cloud Paks。IBM Cloud Paks 以容器方式提供面向企业的开发者工具、数据、AI 服务及开源中间件软件，在 Red Hat OpenShift Cloud Platform 上运行。

此外，IBM 还提供 IBM Cloud Hyper Protect Services，这是构建于 IBM Secure Service Container 技术之上的公有云服务组合，可助开发者轻松使用高度敏感数据构建应用。这些服务包括：

- **IBM Cloud Hyper Protect Crypto Services**：支持用户通过客户控制的硬件安全模块，保管自己的云数据加密密钥。
- **IBM Cloud Hyper Protect DBaaS**：提供极为安全、便捷易用的企业云数据库环境，确保数据在公有云中滴水不漏的保密性。
- **IBM Cloud Hyper Protect Virtual Servers**：通过客户管理的虚拟服务器为敏感工作负载提供完整权限控制。

即使通过不可变基础架构与集中式映像存储库等功能可带来诸多安全优势，容器仍然属于堆栈中需要处理的新层，对于采用者而言，安全性仍是一项重大挑战。LinuxONE 提供诸多可供容器使用的安全功能。LinuxONE 平台的整体设计旨在确保软硬件万无一失。IBM 加密协处理器适配器配备符合 FIPS 140-2 4 级标准的 HSM，通过了 Linux 最高等级安全认证。此外，LinuxONE 还具备 IBM Secure Service Containers 功能，这是一种高度安全的逻辑分区，可为容器化应用打造安全可靠的应用执行环境：

- 内存严密隔离，固件防篡改，启动顺序可信可靠
- 管理员访问权限严格控制，有助于规避特权凭证滥用风险，为使用中数据保驾护航
- 自动以透明方式对所有静态和动态数据进行加密，无任何开销
- 可靠可信，向上扩展能力惊艳
- 与现有数据/应用并置

应用可移植性

过去，由于优化、配置和调优原因，以及选定的中间件与数据管理软件产品套件，应用不仅仅受限于特定操作系统，甚至受限于特定服务器上的某个特定操作系统。随着在 x86 服务器上广泛使用虚拟化功能，操作系统、应用以及所有相关依赖项等整个堆栈从一台虚拟服务器转移至其他虚拟服务器的可移植性更上一层楼。虽然这是向现代计算环境发展的重要一步，但应用最终需要从特定操作系统实例中抽象出来。

要成功实现这类抽象，就需要从底层操作系统的特定实例中，将应用及其依赖项一同抽象出来。若能成功实现这一目标，便能降低与“每个工作负载使用一个虚拟机与一个操作系统”的部署模式相关的开销。

采用解释语言，通过 JavaScript、Perl 和 Python 等通用语言选项，以及 Java 等字节代码编译语言，可进一步加强平台独立性。解释语言可由所运行的平台实时解释，因此开发人员不必担心平台的字节序差异。因此，这样便解决 LinuxONE 系列产品（大端模式）和 x86 服务器（小端模式）的平台差异化问题。

通过对比，广泛使用的编译语言，诸如 C、C++、Go 以及 Haskell，在执行前会进行预编译。这意味着编译器会为要部署应用的平台创建二进制代码。因此，x86 Linux 平台编译的代码无法在 IBM LinuxONE 系统上运行，甚至无法在 OpenShift Container Platform 之类的软件云平台上运行。

为解决这个问题，大多数现代持续集成/持续交付系统都可以管理从通用源代码编译而来的多组二进制文件，并向合适平台部署正确的二进制文件。随着面向 Linux 的跨平台编译器的推出，在 x86 Linux 系统上编译 Z Linux 二进制文件成为现实。此外，OCI 容器映像还具备多架构功能，单个映像可包含多种系统的二进制代码，无需使用多个映像。

未来展望

行业方向

信息技术有着超速发展的趋势，至少对于激动人心的新兴技术，业界满怀热情，积极采用。而实际上，任何技术都拥有可能延续数十年的支持能力。因此，客户应高瞻远瞩，寻求采用具备可移植性、灵活性和可支持性的最佳长期方案。

由于业界已将 Kubernetes 定为容器控制平台，Kubernetes 成为了创新集成纽带，可为业界创造多重价值。首先，Kubernetes 本身可从着力单一平台的庞大社区中受益。其次，创新型项目层出不穷，虽然它们本身不属于 Kubernetes，但仍集成于 Kubernetes，并为其而优化。

比如，Istio 服务网络以及面向无服务器计算的 Knative 可直接在 Kubernetes 技术上进行构建。这些项目旨在培育统一的大型社区，并将成为 Kubernetes 部署的重要组成部分，打造一个更加广泛使用的容器平台。随着用户对混合云和多云的依赖性不断增强，拥有更多跨各种 Kubernetes 发行版本和云服务通用的组件可为他们带来诸多裨益。

IBM LinuxONE 通过多种方式直接参与社区建设，包括：

- **Red Hat OpenShift Container Platform on IBM LinuxONE：**在 Red Hat 产品服务组合中，IBM 最为青睐的便是 OpenShift Kubernetes Platform。这项技术可作为混合云的基础，能够扩展多个超大型云环境，包括 Amazon Web Services、Google Cloud Platform 以及 Microsoft Azure，并为 Istio 与 Knative 提供支持。对 OpenShift 的支持已经整合到 Red Hat Enterprise Linux 支持中，后者已在 LinuxONE 服务器上提供多年。
- **IBM Cloud Paks on LinuxONE：**8 月初，IBM 发布了面向 OpenShift 的 Cloud Paks 概念。这些容器化产品服务包含各类产品，旨在支持数据与人工智能 (Cloud Pak for Data)、应用现代化与云原生应用开发 (Cloud Pak for Applications)、企业应用集成 (Cloud Pak for Integration)、业务流程与决策自动化 (Cloud Pak for Automation) 以及多云管理 (Cloud Pak for Multicloud Management)。
- **IBM Cloud Private on LinuxONE：**在收购 Red Hat 后，IBM 产品服务组合中新增了许多新技术。IBM 现有的 IBM Cloud Private 技术将与这些新技术一同得到 IBM 的支持与投资保护。
- **Linux 发行版：**它们可用于 LinuxONE，包括 Canonical Ubuntu、Red Hat Enterprise Linux 和 SUSE Linux Enterprise Server。此外，还提供了多个开源社区发行版本，以便有需要的客户使用它们来处理非关键工作负载。不仅如此，容器也可以使用诸如 Alpine 以及 Red Hat Enterprise Linux CoreOS 等轻量级内核发行版本。

挑战/机遇

挑战：容器可通过开发者抽象出大量内容，无论底层系统如何，开发者均可通过相同 API，以同样的方式自如工作。然而，底层系统仍会对容器执行、扩展以及安全方式产生重要影响。系统供应商目前的挑战在于将这些差别与优势传达给那些将基础架构视为商品的受众。

商机：通过容器，云原生开发者之类的现代新受众可以轻松使用 IBM LinuxONE 系统。这些开发者借助容器，可以采用处理任何其他基础架构的方式，在 LinuxONE 系统上自如开发，自在部署。这样就可将 LinuxONE 开发整合到现代工作流之中，使之参与容器部署，而开发者无需专门接受有关该平台的培训。然而，对于 Z 系统的操作人员而言，由于他们需要处理容器平台的部署和集成任务，因此还是存在一定的学习难度。

挑战：多数开发者拥有丰富的 x86 环境开发经验， 而对其他架构的开发经验则相对欠缺。

商机：解释语言的移植性更佳；编译语言对于多编译情景、跨平台编译器以及多架构容器的支持性能日益提升。如今，多种非 x86 处理器的运用，包括 GPU、ASIC、为人工智能而优化的自定义处理器、ARM 处理器以及边缘的 Raspberry Pi 等其他技术，意味着开发者正在借助工具为共同配备 Kubernetes 的异构部署环境提供支持。

挑战：IBM Z 被归入大型机“传统”部分。

商机：IBM LinuxONE 具备远超商用平台的灵活性、安全性与可靠性，是多云部署重大商机中非常有价值的一个方面。此外，LinuxONE 生态系统不断发展壮大，数以百计的开源工具与软件已可供使用。虽然在很大程度上，开发者对配备容器的底层系统一无所知且无需特殊技能，但基于多种需求，了解这些功能对于了解容器中可能包含的应用类型仍然至关重要。担当站点可靠性工程师角色的系统供应商与客户所面临的挑战在于，如何通过容器平台凸显底层系统的价值和优势 — IBM LinuxONE 能够根据所需的功能选择适合的容器。

挑战：设法转投远程解决方案的客户不会想在自己的数据中心内构建大型系统。

商机：IBM 提供由 LinuxONE 支持的 IBM Cloud Hyper Protect Services，在 IBM Cloud 中运行。IBM 全新的 LinuxONE III 系统采用行业标准的 19 英寸机架，可轻松安装在标准数据中心内。

结论

如今的时代，创新飞速发展，需求不断增加，IT 行业高管需要制定切实可行的混合云与多云架构方案，满足其特定应用需求，并为混合云环境的其他维度提供扩展性、安全性和可移植性。随着基础架构与部署软件层标准化，客户能够也应该将更多自身资源集中用于打造应用的差异性，包括优化用户体验，加强对多个用户设备的支持，完善应用自身功能。当然，这里假设差异还包括可靠性、可扩展性、安全性和运营成本等核心特性。

IBM 为业界带来了不同凡响的创新平台，为当今的开发者提供了许多亟需的软件和部署经验以及特性。这些服务大多从底层平台抽象出来，尤其是通过 Red Hat OpenShift Container Platform 以及 IBM 平台上的一系列安全可靠、灵活扩展的容器部署服务提供了支持，向寻求行为属性与 IT 投资回报的更广泛潜在受众全面深入展现了 IBM LinuxONE 系统的价值。

关于 IDC

International Data Corporation (IDC) 是全球首屈一指的信息技术、电信和消费科技市场情报、咨询和活动服务供应商。IDC 致力于帮助 IT 专业人士、业务高管和投资机构以事实为基础，做出有关技术采购的决策，制定业务发展战略。IDC 在全球拥有超过 1,100 名分析师，他们从全球、区域和本地视角对 110 多个国家或地区的技术与行业机会和趋势提供专业化的指导意见。50 多年来，IDC 一直为客户提供战略洞察，帮助他们实现关键的业务目标。IDC 是 IDG 旗下子公司，IDG 是全球领先的技术、媒体、研究及活动服务公司。

全球总部

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter：@IDC
idc-community.com
www.idc.com