文章编号:1001-9081(2025)S1-0158-05 DOI:10.11772/j.issn.1001-9081.2024040483

自主创新异构环境容器云平台的设计与适配

吕依濛*
(中国煤炭科工集团 中煤科工开采研究院有限公司,北京 100013)
(*通信作者电子邮箱lvyimeng@live.com)

摘　要:Kubernetes是一种开源的容器管理技术,通过灵活的应用程序编程接口(API)和工具,实现对容器化应用程序的动态扩展,是目前主流的容器管理技术之一。针对基于Kubernetes容器云平台在国产环境下的设计与适配过程,总结容器技术在当前自主创新产业背景下的一些可行性设计方案与解决思路。首先,介绍平台针对国产环境的架构设计方案;其次,从适配角度介绍平台功能在国产化环境的性能及调优;最后,提供具体的解决方案和实例,旨在为后续容器云领域的国产化适配提供参考。平台的设计和适配研究为构建自主创新环境下的容器云平台提供了经验和解决方案,特别是在国产化软硬件环境中的适配和部署方面,提供了具体的适配参数和技术参考。

关键词:Kubernetes;容器;云平台;自主创新产业;国产化适配
中图分类号:TP319　　**文献标志码**:A

Design and adaptation of container cloud platform for independent innovation and heterogeneous environment

LYU Yimeng*
(*CCTEG Coal Mining Research Institute Company Limited*, *China Coal Technology and Engineering Group*, *Beijing* 100013, *China*)

Abstract: As an open-source container management technology, through flexible Application Programming Interfaces (APIs) and tools, Kubernetes enables the dynamic expansion of containerized applications. It is one of the current mainstream container management technologies. Some feasible design schemes and solution approaches for container technology in the current independent innovative industry environment were summarized, focusing on design and adaptation process of the Kubernetes container cloud platform in a domestic environment. Firstly, architecture design scheme of the platform tailored for the domestic environment was introduced. Then, from the adaptation perspective, performance and tuning of the platform functions in a domestic environment were discussed. Finally, specific solutions and examples were provided to offer references for future localization adaptations in the container cloud domain. The above research provides valuable experiences and solutions for building a container cloud platform in an independent innovation environment. In particular, it offers specific adaptation parameters and technical reference for adaptation and deployment in a domestic software and hardware environment.

Key words: Kubernetes; container; cloud platform; independent innovation industry; localization adaptation

0　引言

Kubernetes是一种基于开源代码的容器管理技术,提供对Docker容器的基本管理功能,实现从代码编译、镜像生成到自动化测试及部署的流水线管理,是目前主流的容器管理技术之一。Kubernetes配合Docker技术提供了一种跨平台方式部署和管理容器化应用程序,通过灵活的应用程序编程接口(Application Programming Interface, API)和工具,实现动态扩展,丰富了分布式组件和微服务治理功能,作为管理规模化容器应用的强大工具,在各个公司的生产环境中被广泛使用,中煤科工开采研究院采用这种技术建设了国产化容器云平台,作为公司微服务架构的基础,实现负载均衡、弹性伸缩等功能。

然而,Kubernetes本身技术复杂,建设基础设施需要投入较高成本。本文以Kubernetes为基础,讨论中煤科工开采研究院容器云平台的建设,从前期的架构设计和技术选型,到中期基于国产化软硬件的开发和平台设计,再到后期在国产环境下云平台及容器应用的部署和适配,积累了宝贵的经验。同时,研究了以Kubernetes为核心技术的容器云平台建设与适配,并总结了笔者在自主创新环境下建设平台的实践经验。

此外,在国有企业国产化软件替代过程中,软件适配是一个非常重要且长期持续的工作。如今,自主创新产业中,众多国产芯片、操作系统、数据库及中间件技术蓬勃发展,Kubernetes凭借出色的跨架构管理能力,更值得关注。容器服务与传统基于操作系统的基础服务一样,在虚拟化技术领域中,国产数据、中间件和应用软件适配竞争激烈。同样,在容器云平台上进行相关第三方容器应用的国产化部署及适配工作中也有不少经验和教训值得借鉴。

因此,本文从容器云平台在国产化软硬件环境下设计部署和第三方容器应用在国产化部署适配这2个主要方向论述。针对容器云平台国产化建设及适配的一些具体实例和问题,列出相应的解决方案,为后续容器云领域的国产化适配提供具体的解决方案。

1　平台设计

容器云平台在设计、开发、适配和部署过程中积累了许多关于架构和配置的实践经验,这些经验为Kubernetes容器技术在国产环境下的架构部署及技术选型提供了参考价值。

1.1　数据存储

容器云平台在运行期间产生和使用的数据可依据数据类型、安全等级、实际需求,灵活存储在本地存储和外接存储中。对于外接存储,国内访问Amazon S3、Google Cloud等境外存储服务的速度较低,并且不符合自主创新产业对存储服务的硬性要求。即使国内如百度、华为等厂商提供外接存储服务,目前也没有针对它们的安全性和可控性进行相关标准的认证。经研究对比,相较于使用第三方提供的公有存储服务,若条件允许,在国产硬件环境下搭建自我运维的私有外接存储设备,并选择适合的存储技术,是更符合自主创新产业标准的合理选择。

同时,Kubernetes本身内置了多种本地存储卷类型,例如EmptyDir、HostPath和Local Persistent Volume等,以满足Pod运行时对数据的快速访问需求或高性能应用场景的要求。这些存储

收稿日期:2024-04-23;**修回日期**:2024-10-15;**录用日期**:2024-10-29。
作者简介:吕依濛(1[redacted]),男,北京人,硕士,主要研究方向:容器技术、大数据分析、人工智能。

卷类型经过国产环境的适配，能够在容器云平台上实现Pod与本地存储介质之间高效、安全的访问，从而形成持久化存储。此外，容器云平台对关系型数据也有持久化存储的需求。相较于MariaDB、MySQL和Oracle等传统关系型数据库，更倾向使用国产数据库，例如人大金仓、武汉达梦、南大通用，以及新兴的巨杉、PingCAP等。对此，容器云平台在关系型数据库选型阶段，分别对几种主流的关系型数据库进行国产环境适配，并对这些关系型数据库进行目前业界公认的联机事务处理性能测试C（Transaction Processing performance Council benchmark C，TPC-C）基准测试。以tpmC（每分钟内系统处理的新订单数）值为参考，比较几个数据库在容器云平台上的事务处理能力，其中人大金仓的KES V8性能表现最突出。容器云平台以此为依据，选用人大金仓产品为平台默认内置数据库，同时兼容武汉达梦和南大通用，作为备选。表1为容器云平台在2种存储介质上金仓数据库的TPC-C基准测试结果。

1）本地存储是直连服务器的存储设备，由硬盘或闪存驱动器组成。本地存储提供了高性能的数据访问速度，但容量受物理限制。这种存储方式适合需要频繁读写数据的应用。为了保证关系数据库的性能，在平台适配过程中，对本地存储卷和网络附加存储（Network Attached Storage，NAS）中金仓数据库的性能进行TPC-C基准对比测试。参考表1，金仓数据库在本地存储卷中的性能远高于在NAS存储中的表现。因此，容器云平台的关系数据库系统的架构设计方案选择部署在本地存储，以保证数据库的性能。

2）外接存储是通过网络连接服务器的存储设备，适用于需要数据持久化存储和共享的应用场景。容器云平台使用的外接存储为海康威视的海康存储，具备自主知识产权，采用国产生产线生产，符合自主创新产业对存储硬件的基本要求。根据截至目前容器云平台的运行表现，数据访问性能良好，虽然偶尔出现挂载磁盘丢失的现象，但经过运维人员的及时处理，能够迅速恢复磁盘访问，未出现数据丢失且无法恢复的情况。经实践验证，海康存储是现阶段可选择的国产外接存储替代方案之一。

表1　容器云平台在2种存储介质上金仓数据库的TPC-C基准测试结果

存储方式	仓库数	终端数	测试时长/min	tpmC/10^4
本地存储	100	200	10	129
NAS存储	100	200	10	93

架构方面，容器云平台的外接存储方案分为网络存储和分布式存储。

1）网络存储。容器云平台提供了多种网络存储方式，其中以NAS和互联网小型计算机系统接口（internet Small Computer System Interface，iSCSI）为主。

NAS存储通常采用独立磁盘冗余阵列（Redundant Array of Independent Disks，RAID）技术实现数据冗余和数据备份，访问速度虽不及本地存储和iSCSI，但是保证了数据的高可靠性和数据安全性，适合需要大容量存储的应用，如文件共享、多媒体存储和备份。容器云平台采用NAS协议为默认存储方式，用于存储平台和不同容器之间产生的非关系型数据，这些数据具有数据量大、访问浏览多余修改更新等特点。参考实际平台运行情况，NAS存储是目前国产环境下，基于大数据量首选的网络存储方案之一。

容器云平台同样兼容iSCSI协议。iSCSI存储具有连接服务器数量无上限、在线扩容、动态扩容等特点，虽然在国产环境和大数据量的场景下，总体性能不及NAS存储，但对于存储集成或灾难恢复的应用场景，iSCSI存储的性能明显优于NAS存储[1]。

容器云平台根据实际应用场景灵活调整网络存储协议，以满足不同客户对数据访问性能和安全性的要求。

2）分布式存储。容器云平台提供多种分布式存储方式，包括网络文件系统（Network File System，NFS）、Ceph、GlusterFS（Gluster File System）、开源企业级块存储（Open Enterprise Block Storage，OpenEBS）等。

分布式存储用于高可用和高性能的应用场景，容器云平台部署的几个基于国产芯片架构的高可用集群，例如C86、高级精简指令集机器（Advanced RISC Machine，ARM）、LoongArch和SW64，都是采用分布式存储作为外接存储的选型方案。

从整体存储架构方案的角度，容器云平台充分结合了本地存储卷和外接存储的优点进行分层部署。不同的微服务模块根据各自的需求和权限访问不同的存储，例如，数据库或虚拟化环境可以使用本地存储卷，以满足高性能数据访问的需求；而文件共享和多媒体存储则可以使用NAS存储，从而实现大容量数据存储和共享。

使用这种架构方案既可以保证存储访问的高容量和高性能，也可以提高数据的安全性和可靠性[2]。

1.2　网络配置

Kubernetes使用Calico或Flannel等网络组件。容器云平台的网络配置方案灵活，可以根据用户的实际业务场景选择不同的网络方案实现不同的功能。

平台结合Kubernetes网络组件，采用以下4种配置方案：

1）Cluster内部网络。这种网络用于容器之间的通信，以及Pod和Service之间的通信。一般使用基于互联网协议（Internet Protocol，IP）的网络方案，如Flannel、Calico、Cilium等。其中，Flannel通过配置节点之间的虚拟网络实现容器之间的通信；Calico使用边界网关协议（Border Gateway Protocol，BGP）实现路由和安全策略；Cilium则结合伯克利包过滤器（Berkeley Packet Filter，BPF）技术提供网络和安全功能。

2）Service访问网络。这种网络用于外部服务访问容器云平台中的Service，平台使用ClusterIP和LoadBalancer实现容器访问。ClusterIP为每个Service分配一个虚拟IP，平台用户可以通过虚拟IP访问对应的Service。LoadBalancer则通过负载均衡调节器集中管理，将流量转发到Service上。目前平台集群中，C86、ARM架构集群通过LoadBalancer集中管理，而LoongArch、申威64（ShenWei 64，SW64）、无互锁流水线的微处理器（Microprocessor without Interlocked Pipeline Stages，MIPS）架构则通过ClusterIP方式实现IP访问。

3）Ingress网络。这种网络用于将外部流量路由到平台中的Service，通过配置Ingress Controller，可以实现对外部流量的负载均衡、安全套接字层（Secure Sockets Layer，SSL）终止、路径匹配等功能。

4）存储网络。这种网络用于Pod访问持久化存储，结合2.1节的外接网络存储方案网络存储协议和分布式存储协议实现。另外，平台还提供容器存储接口（Container Storage Interface，CSI）规范，可以通过CSI插件实现存储网络的集成[3]。

由于网络连通性和带宽等因素的限制，容器云平台主要采用本地化的网络方案，并灵活部署本地化的软件定义网络（Software-Defined Networking，SDN）方案，形成4种方案的混合架构。该架构方案有效保障了容器云平台的网络性能，使容器云能够在国产架构环境下以最佳性能状态运行，同时也在性能和安全性之间达到平衡[4]。

1.3　信息安全

容器云平台从前期设计开发到后期适配部署中存在安全性问题，尤其是在国产环境下，基于容器的信息安全问题较为严峻。以下是容器云平台在国产环境下开发适配时，结合实际经验，总结的5点需集中关注的安全问题：

1）容器漏洞。由于容器是基于操作系统的虚拟化技术，所以容器中的操作系统和软件也可能存在漏洞。国产环境下，芯片如LoongArch、SW64等尚处于自主创新的起步阶段，操作系统如麒麟、统信等商业发行版尚未脱离Ubuntu发行版的窠臼，整个国产化体系架构在很多领域中向自主可控的目标持续迈进。因此，国产环境下未发现、易攻击的漏洞相较于传统环境更多。攻击者可以通过利用这些漏洞轻易攻击容器内的应用程序和数据。

2）镜像安全。容器使用镜像是应用程序的基础，而这些镜像可能包含安全漏洞。当容器镜像运行在国产中央处理器（Central Processing Unit，CPU）架构时，如LoongArch、MIPS、SW64等，相较于X86体系，不可预知的安全漏洞更多。攻击者可以通

过篡改镜像在应用程序中植入木马病毒等恶意代码，非法操纵容器环境，危害镜像安全。

3）数据隔离。多个容器共享同一台物理主机，因此需要确保容器之间的数据隔离。如果没有适当的隔离措施，攻击者可以通过一个容器中的漏洞访问其他容器中的数据。容器云平台通过跨CPU体系架构的集群管理模式，在保证不同架构的镜像实例能够通过Pod共享数据的同时，也研究了控制容器间数据的隔离安全机制。

4）容器管理安全。容器管理系统需要提供强大的安全管理功能，以确保容器运行环境的安全性和稳定性。例如，需要实施国产密码传输加密、认证、授权和计费（Authentication，Authorization，and Accounting，AAA）等措施。

5）网络安全。容器需要连接网络，因此网络安全也是一个重要的问题。需要结合2.2节的容器云平台网络配置体系，实施网络隔离、流量监测和访问控制等措施，以确保容器环境的安全性[5]。

针对以上几个安全问题，为提高国产环境下的信息安全性，容器云平台采取了以下措施：

1）遵循最佳实践原则：包括限制容器的权限、加强镜像安全、使用最小化的基础操作系统和软件等。容器云平台针对5种主流国产基础架构（C86、ARM、LoongArch、SW64和MIPS），对容器的权限及参数配置都进行定制化的设计和开发。

2）网络隔离和服务网格：通过在不同的命名空间中部署不同的Pod和Service可以实现网络隔离，避免不同镜像实例之间的网络冲突和干扰。服务网格用于管理容器间的网络流量，控制不同组件之间的交互。通过部署服务网格框架，如Istio、Linkerd等，可以在容器云平台网络层面提供更细粒度的服务路由、流量控制和安全功能，保障网络的安全性。平台定制开发了多种适用于国产环境下的安全控制和功能保障原则。

3）安全审计和日志管理：记录和监控容器的活动，及时发现和响应安全事件。通过日志记录、监控等手段审计网络流量，及时发现异常行为，保障网络的安全性。容器云平台采用独立的运维检测系统全方位检测平台微服务模块，实时监测平台运行时的状态及参数，完成日志记录及监控，保障平台在国产环境下的信息安全和信息数据的完整性，随时掌握适配阶段系统实时反馈的有效信息，是平台适配国产化环境的有效手段之一。

4）容器和网络的加密：例如使用支持加密、密钥管理和安全审计的容器。通过配置传输层安全性（Transport Layer Security，TLS）证书、互联网协议安全（Internet Protocol Security，IPSec）等方式，可以对网络进行传输加密，避免敏感信息被窃取。后期计划采用国产椭圆曲线公钥密码算法（Elliptic Curve Public Key Cryptography Algorithm，SM2）算法SSL（Secure Sockets Layer）证书替换现有RSA（Rivest-Shamir-Adleman）非国密算法SSL证书，确保整个国产体系架构在网络传输中的安全性。此外，在网络加密方面，通过软件加密、物理加密机、加密云平台相结合的方式，构建一套国产环境自主可控的密码可信体系，保证容器云平台在网络加密领域的安全[6]。

5）AAA认证：一种确保网络安全的管理机制。容器云平台采用微服务架构，所有微服务模块访问全部通过运管平台统一管理。使用身份验证和授权策略管理容器环境中的访问，同时记录网络服务流量。对于容器云平台的身份验证、授权策略和流量统计，操作对象基于微服务模块层，可以针对不同的微服务进行身份验证、授权策略和流量统计，在整个生态系统的云管平台中集中管理。通过平台内置的基于角色的访问控制（Role-Based Access Control，RBAC）机制或其他网络安全方案，可以控制网络访问，保护敏感数据和应用。

1.4 第三方厂商

在容器云平台的建设过程中有专业容器云应用、数据平台服务提供商和自主创新产业各个领域厂商的技术支持，包括芯片、整机、网络设备、数据存储、固件、操作系统、中间件和信息安全等上下游领域。目前，国外如亚马逊网络服务（Amazon Web Services，AWS）、微软Azure，国内如华为云、阿里云、腾讯云、百度云等云服务厂商，Kubernetes技术已被广泛支持，且功能和性能都在不断地升级迭代。

为了构建自己的容器云平台，提供优质的容器云服务，以获取更好的支持和服务，调研国内多家主流容器云应用服务商和上下游生态合作伙伴。针对功能全面、可扩展性强、兼容性强等核心关注点，并在具有联合建设意愿的背景下，与这些服务提供商共同研发容器云平台。针对诸如容器云平台部署的轻量化、开发运维一体化流水线自动封装、跨架构微服务治理的跨芯片灰度发布等国产环境的核心问题联合研发，提供相应解决方案。

容器云平台在选择服务提供商时主要基于以下原则：

1）通过跟服务提供商的联合建设，能够提供一种基于Kubernetes的跨平台解决方案，兼容X86、ARM、LoongArch、SW64、MIPS等国产主流CPU技术路线，可以在多个云和数据中心中运行。这使得信息技术应用创新产业可以更灵活地部署和管理应用程序，摆脱外国芯片架构的硬性束缚，从而扩展业务和客户基础。

2）通过与服务提供商的联合建设，能够提供企业级容器云技术支持，包括开发运维一体化（Development and Operations，DevOps）、中间件服务、微服务治理、应用性能管理（Application Performance Management，APM）性能监控、分布式服务总线、服务网格和API网关等核心服务功能，能够在国产环境下为用户提供强大的功能和技术支持。

可以说，容器云平台是在同自主创新产业上下游生态服务厂商在联合建设、联合服务、联合运营的理念下，共同设计、研发完成的。在国产环境下建设以容器云平台为代表的各种信息系统时，是一个具有参考价值的典型案例。

2 平台适配

容器云平台在国产化适配阶段，着重在5种主流国产芯片架构（C86、ARM、LoongArch、SW64、MIPS）上进行了大量的国产化适配工作，适配了大量的软件和应用，并在国产环境下研究像弹性伸缩、灰度发布等功能。

2.1 镜像仓库

由于国内网络限制，Docker官方的镜像仓库访问速度较低。为了提高访问速度，可以使用国内的镜像仓库，如阿里云、网易云等。但是，由于国产产业的特殊性，一些主要功能架构和服务模块需要部署在内部网络。因此，从安全性和效率的角度，有必要在内部网络部署私有镜像仓库[7]。

因此，经过国产化适配后，容器云平台的镜像仓库采取了以下2种方案：

1）基于安全性允许的微服务模块，开通外网访问权限，使用国内的公有镜像仓库，如阿里云、腾讯云等，同时使用DaoCloud容器镜像加速器，进一步提高镜像拉取速度。

2）基于安全性部署在内网的微服务模块，容器云平台在内网构建了私有镜像仓库，基于Docker官方提供的DockerRegistry开发，包括依赖管理、版本控制、日志记录等功能，以满足构建一般镜像的基本需求。

容器云平台默认情况下会访问平台内部的Harbor私有镜像仓库以获取资源。如果私有镜像仓库无法提供相应资源，则平台会访问符合国产产业要求的公有镜像仓库获取资源。通过以上2种方案的主备使用，同时确保容器云平台对镜像仓库访问的性能要求和安全要求[8]。

2.2 应用商店

在容器云平台适配软件过程中，发现很多软件都是业界通用的。第一次适配成功后，许多业务场景中都可以复用。例如，Nginx（Engine X）、Tomcat、东方通、金蝶等常用的非国产和国产Web容器中间件应用，几乎每个实际Web端访问的业务服务在制作镜像时都需要使用这些基础镜像。

为高效复用这些已适配国产化的基础应用成果，平台利用Helm技术，将这些基础镜像封装成一键部署安装包，并部署到应用商店，方便客户直接使用。

Helm是Kubernetes的一种镜像包管理器技术，与Linux系统中的高级打包工具（Advanced Package Tool，APT）和黄色狗更新

程序（Yellowdog Updater, Modified, YUM）包管理器类似，将一组Kubernetes资源打包统一管理。平台用户可以在应用商店快速查找、下载和安装所需的镜像包，在平台上实现一键化部署。这是容器云平台上查找、共享和使用镜像服务的便捷方式。

由于Helm在不同系统环境下，相同软件包的参数配置各有不同，不能做到一包通用，容器云平台在构建应用商店应用服务时进行了大量的Helm领域的适配工作。

表2列举了一些容器云平台在应用商店适配的基础软件镜像，包括Nginx、Elasticsearch、Consul、Redis、Kafka、PostgreSQL（Postgres Structured Query Language）、RabbitMQ（Rabbit Message Queue）、Jenkins、InfluxDB（Influx DataBase）、MongoDB（Mongo DataBase）、Tomcat、Etcd、Memcached、Yapi和SonarQube。

表2　应用商店适配的基础软件包

名称	描述	C86	ARM	LoongArch	SW64	MIPS
Nginx	代理服务器	是	是	是	是	是
Elasticsearch	文本数据库	是	是	是	是	是
Consul	键值对数据库	是	是	是	是	是
Redis	键值对数据库	是	是	是	是	是
Kafka	消息服务器	是	是	是	是	是
PostgreSQL	数据库	是	是	是	是	是
RabbitMQ	消息服务器	是	是	是	是	是
Jenkins	集成工具	是	是	是	是	是
InfluxDB	时序数据库	是	是	是	是	是
MongoDB	NoSQL数据库	是	是	是	是	是
Tomcat	Web服务器	是	是	是	是	是
Etcd	文本数据库	是	是	是	是	是
Memcached	键值对数据库	是	是	是	是	是
Yapi	测试工具	是	是	是	是	否
SonarQube	扫描工具	是	是	是	是	否

实践证明，通过应用商店集中管理基础软件和应用软件，复用基础镜像是国产环境下提高容器云平台容器使用效率的方法之一。

2.3　弹性伸缩

在跨架构环境中使用Kubernetes时，需要根据平台指标和策略自动调整平台资源，自动新增和缩减镜像实例数，实现高性能的弹性伸缩。

在研究弹性伸缩在国产环境下的性能时，重点考虑以下几点：

1）伸缩指标：根据几种国产主流芯片架构的性能特点，设定不同的伸缩指标，例如CPU利用率、内存利用率、网络流量等。表3展示了容器云平台在不同CPU架构下的弹性伸缩阈值。这些指标和应用程序利用率相关，在不同的架构环境中度量，使容器实例能够在不同的国产芯片架构下保持最大利用率，发挥容器平台最佳性能。

表3　不同芯片架构下弹性伸缩的阈值　　单位：%

CPU架构	CPU扩展利用率	CPU收缩利用率	内存扩展利用率	内存收缩利用率
C86	90	45	95	50
ARM	90	45	90	40
LoongArch	80	40	80	35
SW64	75	35	80	35
MIPS	70	30	70	30

2）伸缩策略：平台根据不同CPU架构下弹性伸缩的伸缩指标阈值设置自动伸缩策略，包括基于平均负载的水平自动伸缩、基于CPU和内存利用率的垂直自动伸缩等，供用户使用。用户可以根据容器实例实际的负载情况，选择适当的自动伸缩策略[9]。

3）资源配额：在弹性伸缩时，需要将容器实例资源配额与平台不同芯片架构集群实际配额相匹配。根据生产环境中不同芯片架构集群的资源使用情况，平台为各个集群分配资源配额，确保不同架构下镜像实例在扩展时有足够的资源。

4）跨架构伸缩：容器云平台在国产化适配时，参考了不同架构下容器的特点，在保证高可用的前提下，实现了跨架构间容器实例的弹性伸缩。平台实现了几种不同芯片架构集群之间弹性伸缩的统一调配及管理，以及在不同容器之间的负载均衡、故障转移和跨架构管理，同时保证了不同集群间的高可用性。此外，MIPS架构由于自身的指令集问题，在同等条件下，无法同时和其他几种国产架构下的容器一样实现高可用[10]。

根据实际国产环境需求研发的动态弹性伸缩功能，平台发挥了容器实例在国产芯片运行下相对高效的性能。

2.4　灰度发布

灰度发布是指平台在发布镜像应用时，相较于传统的上线发布，采取的一种针对不同客户、不同版本，甚至不同时间的一种智能应用发布方式，可以保证镜像应用发布的平稳性、灵活性和安全性。容器云平台包括A/B测试、金丝雀等多种形式的灰度发布形式，平台用户可以根据容器应用的实际发布状态和需求，灵活选择发布形式。

容器云平台不仅适配了几种主流国产芯片架构，还实现了跨架构之间同容器的自动化灰度发布功能。截至目前，平台实现了从C86到ARM平滑的跨架构灰度发布。平台用户可以根据对镜像应用的不同需求，自定义发布的时间、形式和发布的架构种类，实现了异构容器云平台的跨架构发布管理[11]。

3　设计适配实例

3.1　实例范围

在容器云平台从X86架构基础环境向ARM、LoongArch、SW64等国产芯片架构迁移建设时，由于底层指令集、操作系统、中间件、数据库的不同，在迁移过程中遇到了大量技术问题。这些问题是容器云平台适配国产化环境时遇到的主要挑战，可以说是国产化适配中最耗时、最需要技术支持，且耗费研发、测试成本的主要关注点。

这些问题主要集中在系统在国产化软硬件替代过程中产生的兼容性问题和系统错误，可分为以下几个方面：

1）将Intel、AMD为代表的X86架构芯片替换为ARM、LoongArch、SW64等国产芯片遇到的指令集差异产生的兼容性问题和系统错误。

2）将Windows、MacOS等非国产操作系统替换为麒麟、统信等国产Linux操作系统遇到的操作系统差异产生的兼容性问题和系统错误。

3）将Tomcat、Nginx、Apache等非国产Web服务器替换为金蝶、东方通等国产服务器中间件遇到的系统差异产生的兼容性问题和系统错误。

4）将MySQL、Oracle等非国产数据库管理系统替换为金仓、达梦等国产数据库管理系统遇到的系统差异产生的兼容性问题和系统错误。

5）将其他非国产软件替换为国产软件遇到的兼容性问题和系统错误。

针对以上5个方面，容器云平台在进行国产环境适配时遇到了大量兼容性问题和系统错误，并针对具体问题逐一进行了排查和解决，积累了大量容器云平台国产环境适配的经验。在适配过程中及时记录这些实际问题及相关解决方案，并录入平台的知识库系统，可以为今后的国产适配工作规避歧路。

此外，容器云平台在与自主创新产业上下游厂商联合实验过程中，也遇到了许多适配兼容性问题和系统错误。通过对这些问题的联合研究、排查和解决，同样积累了大量国产化适配的经验。联合实验中的适配基于C86和ARM下的国产环境，集中关注解决容器平台轻量化、易部署、高封装程度等问题，以及平台客户对容器灵活、轻量化的定制需求。此外，通过联合研究，解决了国产化环境下DevOps黑盒封装全自动化、C86和ARM间跨架构灰度发布、弹性伸缩等技术难题。

3.2　实例列举

针对以上所述，列举以下几个国产化适配中遇到的问题及

解决方法：

1）持续集成/持续部署（Continuous Integration / Continuous Deployment，CI/CD）让打包过程分别在X86和ARM节点上执行。

在混合云的改造过程中，流水线上需要对项目的前后台代码进行自动打包发布。采取部署2条流水线服务的方式，分别让2条流水线运行在ARM节点和X86节点上，给予2条流水线不同的标签，用于执行各自的任务。

2）应用实例的弹性伸缩的实现。

由于应用在访问量上升时会导致CPU和内存占用的升高，在平台中加入对应用实例占用节点CPU、内存资源的监控，在内存和CPU占用升高时，调用容器编排管理子系统相关接口，对应用实例数量进行相应调整。

3）Web应用程序归档（Web Application Archive，WAR）包项目混合平台节点发布问题。

WAR包项目基于Tomcat容器进行发布部署，首先准备双平台同版本的Tomcat镜像容器，调整流水线的镜像制作模板，在镜像的自动制作过程中不仅生成ARM平台的镜像，还可以生成X86平台的镜像，并将2个平台镜像使用标签标记不同平台，并合并到相同的Harbor目录下管理。

4）应用的权限管控问题。

利用Kubernetes资源隔离的特性，基于命名空间管控应用权限，关联命名空间与项目组成员，具有该命名空间权限的项目组成员可以查看该命名空间下的应用信息，将应用的创建发布都锁定到用户的命名空间下，以实现不同应用的权限隔离。

5）在X86节点适配过程中，内部域名系统（Domain Name System，DNS）无法正常使用的问题。

内部域名无法使用，可能是CoreDNS组件的问题，需要检查它的Pod是否正常运行，同时修改ConfigMap中的CoreDNS配置参数，更新加载该配置信息。具体参数配置如表4所示。

6）在飞腾服务器上搭建容器云时，出现了域名偶尔无法完全解析的问题。

经过排查发现，容器云平台的域名解析与节点配置相关，由于飞腾服务器自带内部DNS，在使用集群DNS时产生了DNS冲突。通过修改CoreDNS的相关配置可以解决此问题。

表4 参数配置

参数	参数说明
errors	输出错误信息到控制台
ready	全部插件已经加载完成时，将通过end points在8081端口返回HTTP状态200
kubernetes	CoreDNS将根据Kubernetes服务和Pod的IP回复DNS查询
prometheus	是否开启CoreDNS Metrics信息接口，如果配置则开启，接口地址为http://localhost:9153/metrics
forward	任何不在Kubernetes集群内的域名查询将被转发到预定义的解析器（/etc/resolv. conf）cache：启用缓存，30 s TTL
loop	检测简单的转发循环，如果找到循环则停止CoreDNS进程
reload	监听CoreDNS配置，如果配置发生变化则重新加载配置
loadbalance	DNS负载均衡器，默认round_robin

4 结语

本文介绍了中煤科工开采研究院基于Kubernetes技术的容器云平台适配国产化环境的一些设计及适配的经验，为国产环境下基于Kubernetes容器功能的设计、开发及部署及适配提供了一种可行性的方案，在保证容器最佳性能的前提下，实现了容器平台在国产环境下安全平稳的运行。随着容器云平台的迭代与升级，相关研究不断深入，将推动自主创新产业与容器技术新的发展。

参考文献（References）

[1] 路明怀，许阳光. 信创云计算技术下虚拟化网关与网络管理研究[J]. 无线互联科技，2024，21(16)：117-121.

[2] 赵勇. iSCSI透明加解密网关中协议解析研究[J]. 软件导刊，2017，16(8)：182-184.

[3] 闫娟雅. 基于Kubernetes的海量网络数据存储方法研究[J]. 电脑知识与技术，2021，17(27)：28-29.

[4] 李俊俊，董建刚，李坤. 基于Kubernetes的集群节能策略研究[J]. 计算机工程，2024，50(9)：82-91.

[5] 王浩硕，李雨含，何亮忠. 基于网络安全网格的云原生安全架构研究与实践[J]. 通讯世界，2024，31(4)：25-27.

[6] 冯兴华. 如何看待容器的安全性[J]. 计算机与网络，2015，41(20)：48-49.

[7] 程宁. 基于Harbor实现Docker私有仓库搭建的研究[J]. 现代工业经济和信息化，2022，12(9)：73-75.

[8] 王伟军. 基于Kubernetes的容器云平台建设[J]. 电脑知识与技术，2019，15(36)：47-48.

[9] 沐磊，李洪赭，李赛飞. 一种改进的Kubernetes弹性伸缩策略[J]. 计算机与数字工程，2022，50(2)：327-331.

[10] 杨武. 混合云平台的设计及实现[J]. 电脑知识与技术，2021，17(11)：77-78.

[11] 杜磊，王竞争，穆启鹏. 基于异构国产CPU的容器应用统一构建系统研究[J]. 电脑知识与技术，2021，17(31)：4-6.