

# 基于 Kubernetes 的卫星遥感数据容器云平台

刘宏娟<sup>1,2</sup>, 黄 炜<sup>2</sup>, 李文吉<sup>2</sup>, 邵治涛<sup>2</sup>, 胡 倩<sup>2</sup>

(1. 自然资源部航空地球物理与遥感地质重点实验室, 北京 100083;

2. 中国自然资源航空物探遥感中心, 北京 100083)

**摘要:** 随着卫星遥感行业进入大数据时代, 传统的数据处理平台日渐难以满足卫星遥感数据快速积累的发展需求, 同时也对系统在动态调度资源、易维护、弹性扩展、复杂度可控和灵活部署等方面提出了更高的要求。针对上述问题, 文章引入容器技术及 Kubernetes 容器集群管理系统, 设计实现了基于 Kubernetes 的开源容器云平台, 将单个业务系统拆分成多个独立运行在相互隔离容器中的服务, 实现了应用服务容器的调度和管理、快速部署和迁移, 并以 GF7 高分卫星数据为例进行验证, 验证了容器集群能够有效处理卫星遥感数据, 提高了卫星遥感数据处理平台的资源利用率和运维效率; 文末结合卫星遥感行业的现状及特点, 就容器化应用的前景及面临的困难进行了展望分析。

**关键词:** 容器技术; 虚拟化; Kubernetes; 软件架构; 卫星遥感

## A Kubernetes based Container Cloud-Platform of Satellite Remote Sensing Data

LIU HongJuan<sup>1,2</sup>, HUANG Wei<sup>2</sup>, LI Wenji<sup>2</sup>, SHAO Zhitao<sup>2</sup>, HU Qian<sup>2</sup>

(1. Key laboratory of airborne geophysics and remote sensing geology, Ministry of nature and resources, Beijing 100083, China; 2. China Aero Geophysical Survey & Remote Sensing Center for Land and Resources, Beijing 100083, China)

**Abstract:** As satellite remote sensing industry enters big data era, traditional data processing platforms are gradually difficult to meet the rapid increase of satellite remote sensing data. It also requires better performance of the system on dynamic resource scheduling, maintenance, extendibility and deployment, etc. These features happen to be the advantages of the open source framework Kubernetes. Therefore, a Kubernetes based container cloud-platform of satellite remote sensing data was developed, spreading an individual business system into multiple independent running services into mutual isolation containers. The platform realizes the scheduling and management of application service container, rapid deployment and migration, and improves the resource utilization and operation and maintenance efficiency. Besides, this article analyzes the prospects and difficulties of container applications combined the feature of satellite remote sensing industry.

**Keywords:** container technology; virtualization; Kubernetes; software architecture; satellite remote sensing

## 0 引言

卫星遥感数据的处理及应用需要强大的算力支持, 目前在用的数据处理平台以单体服务器和虚拟化平台为主。随着我国航天事业的高速发展, 卫星遥感也进入了“大数据”时代, 一方面, 卫星遥感行业按照地质调查整体规划要向着一张图、一站式、平台化

的方向发展。另一方面, 卫星遥感数据的显著特点是多元异构, 除了基本的卫星影像之外还包含了时间、空间、频谱等复杂的地质属性信息, 卫星遥感的数据类型日渐丰富, 数据总量呈现海量增长趋势, 对硬件资源动态分配、弹性扩展、快速部署等方面均提出了更高要求。因此, 探索利用云计算、大数据等新技术构建更加高效便捷的数据处理平台近年来在卫星遥感

收稿日期: 2021-10-22; 修回日期: 2021-11-26。

基金项目: 航遥青年创新基金(2020YFL31); 中国地质调查局专项(DD20191006)。

作者简介: 刘宏娟(1989-), 女, 河南方城人, 研究生, 工程师, 主要从事大数据, 云计算, 信息系统建设方向的研究。

通讯作者: 黄 炜(1982-), 男, 河南郑州人, 高级工程师, 大学本科, 主要从事信息化平台建设, 卫星业务系统规划方向的研究。

引用格式: 刘宏娟, 黄 炜, 李文吉, 等. 基于 Kubernetes 的卫星遥感数据容器云平台[J]. 计算机测量与控制, 2022, 30(1): 209-214, 220.

行业愈发重要, 并已在地质调查的其他领域内取得了进展<sup>[1-2]</sup>。本文引入了容器概念以及 Kubernetes 容器集群管理系统, 以容器为基本单位代替虚拟机部署应用服务, 充分发挥容器轻便快捷、弹性扩展、便于维护的优势, 尝试构建符合卫星遥感行业特点的容器云平台<sup>[3]</sup>。利用 Kubernetes 对容器的调度管理机制, 将计算、网络、存储等服务器硬件资源虚拟化后动态分配给容器, 为运行在容器内的卫星遥感应用服务提供高效可靠、轻便统一的“一站式”服务。

## 1 容器与虚拟化

### 1.1 容器技术

容器 (Container) 技术是一种轻量级的操作系统层虚拟化技术, 借鉴了远洋运输里“集装箱”的理念。一般而言, 部署一个应用需要提前准备好物理主机、操作系统以及各种中间件等软硬件环境, 虚拟化部署一定程度上实现了硬件资源复用, 但没有解决应用层问题, 针对每一项应用部署还是需要单独搭建一整套依赖的操作系统、应用环境等。容器技术的原理是基于 Linux 的 Namespace 和 Cgroup 机制, 将应用打包并虚拟化封装出了一套内核轻量级的操作系统 (NameSpace 区分隔离容器, Cgroup 管理控制系统资源)<sup>[4]</sup>, 通过容器引擎共享操作系统, 只需在容器内部打包必要的 Lib/Bin 等操作系统文件, 具有极其轻量、秒级部署、易于移植、弹性伸缩等特点, 极大提高了部署和运维效率<sup>[5]</sup>。

Docker 是一个基于 LXC (Linux Container) 的开源容器引擎, 将与应用相关的依赖项 (包括环境、程序、文件) 打包成镜像部署在物理机中, 主要包括容器 (Container)、镜像 (Image) 以及仓库 (Repository) 3 个部分。由于容器是应用程序的抽象, 将各类应用服务变成模块化的组件, 容器可以在相互隔离的物理环境中独立运行, 物理机上可以同时运行多个容器, 容器之间共享操作系统。利用镜像仓库, 运维人员可以管理不同的镜像版本, 灵活配置、快速部署。由于 Docker 标准化、轻量级、易迁移扩展的优势, Docker 已发展成了容器技术的代名词, 是目前应用最广泛的容器引擎<sup>[6]</sup>。

### 1.2 容器与虚拟化

虚拟化能更好地利用硬件资源, 同一台物理主机上可以部署运行多个虚拟机, 虚拟机之间资源和进程彼此隔离, 解决了传统模式下同一台物理机上的应用服务抢夺资源问题。虚拟机是在物理主机中虚拟出一

台完整计算机, 在虚拟化的硬件之上运行所有组件 (包括自身操作系统), 而容器则是直接将物理主机的操作系统虚拟出独立的用户空间, 每个用户空间分配有各自的 CPU、内存等计算资源供容器使用, 容器之间彼此独立但共享物理主机的操作系统<sup>[7]</sup>。因而, 容器的启动更迅速, 创建更加简便快捷, 能支持大规模应用服务的快速部署更新。虚拟化与容器的差异对比如图 1 所示。

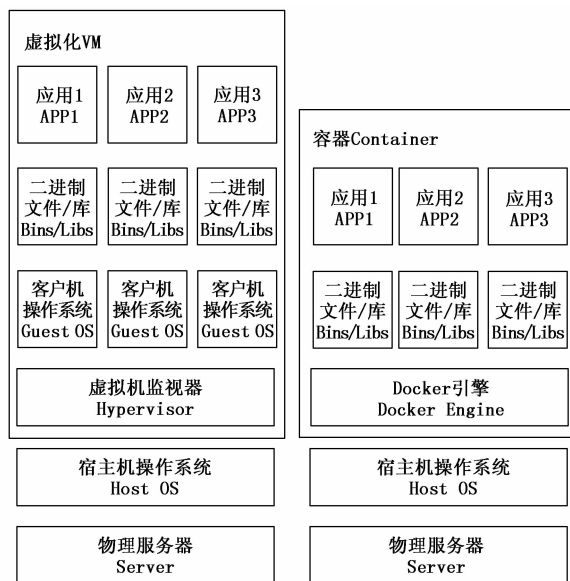


图 1 虚拟化与容器对比

### 1.3 Kubernetes 核心概念

随着容器化部署的应用不断增多, 容器和物理主机的集群规模也随之不断刷新, Docker 逐渐无法满足大规模容器集群的管理需求。Kubernetes 是 Google 公司以 Docker 为基础、面向无服务场景的开源容器集群管理系统, 能够为容器化的应用服务提供资源调度、自动重启、负载均衡、服务发现等基础功能。Kubernetes 支持跨物理主机管理容器, 大大降低了容器平台的维护难度, 同时还为开发人员提供了一套 RESTful 接口, 为后续开发拓展打下了良好基础。鉴于 Kubernetes 强大的开源社区支撑以及众多成功应用的案例, 是当下较为主流的容器管理平台<sup>[8-9]</sup>。

## 2 系统设计

### 2.1 系统整体框架

基于 Kubernetes 卫星遥感数据容器云平台的逻辑功能框图如图 2 所示, 包括物理层、Kubernetes 平台层和应用层 3 部分<sup>[10-11]</sup>。为保障卫星遥感业务

正常运行, 容器云平台之外还需要配套相应的支撑服务。本文选择了两个典型支撑服务作为代表: 分布式存储和 ES (Elastic Search) 集群。

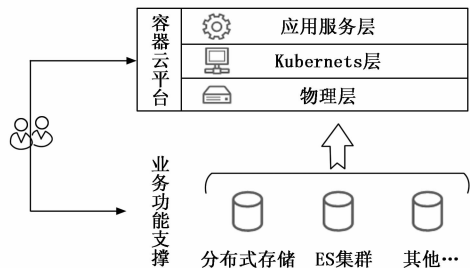


图 2 容器云平台逻辑架构

物理层主要是指由实体服务器池化虚拟出来的物理资源 (包括计算资源、存储资源、网络资源), 这些物理资源会被上层的 Kubernetes 调度并分配给不同的容器。Kubernetes 层是整个容器云平台的核⼼, 统一管理控制容器及应用层服务的生命周期、资源分配, 实时监控管理容器及部署在容器中的应用服务<sup>[12]</sup>。应用层主要是将卫星遥感应用软件抽象成一个个服务并容器化地部署到容器中, 每个容器拥有独立的物理资源, 容器彼此之间共享物理主机的操作系统, 构成了针对特定应用服务的轻量内核级“虚拟机”<sup>[13]</sup>。

对于业务支撑功能, 分布式存储系统为容器云平台提供统一的外置存储服务, 主要用于海量卫星遥感数据的存储、访问、备份, 既简化了应用层的数据存储复杂度, 也保证了数据的一致性、可靠性。ElasticSearch 为卫星遥感应用服务提供数据检索服务, 其索引能力强大, 且支持多种查询类型, 在地理信息数据索引查询方面优势明显。ES 集群结合卫星遥感数据特点进行了单独优化设计, 能够大幅提高检索效率。由于上层应用服务需要频繁地检索使用卫星遥感数据, 因此将检索功能独立部署于容器云平台之外, 通过接口提供服务。容器云平台使用多种接口协议进行访问: ES Rest API、ES SQL、Impala SQL、Hive SQL、HBase MapReduce, 也充分体现了平台架构的开放性。

## 2.2 Kubernetes 架构

Kubernetes 主要由控制节点 (Master node) 和工作节点 (Worker node) 两类节点组成, 其系统架构如图 3 所示。<sup>[14]</sup>控制节点包括 API 服务器、控制管理器、调度器和 etcd 键值数据库等组件, 负责全局决策、调度控制、事件响应等平台管理功能。<sup>[15-16]</sup>

工作节点主要包括 kubelet 和 kube-proxy 两个组件, 用以维护工作节点上运行的每个 Pod, 并为每个节点提供必要的运行环境。

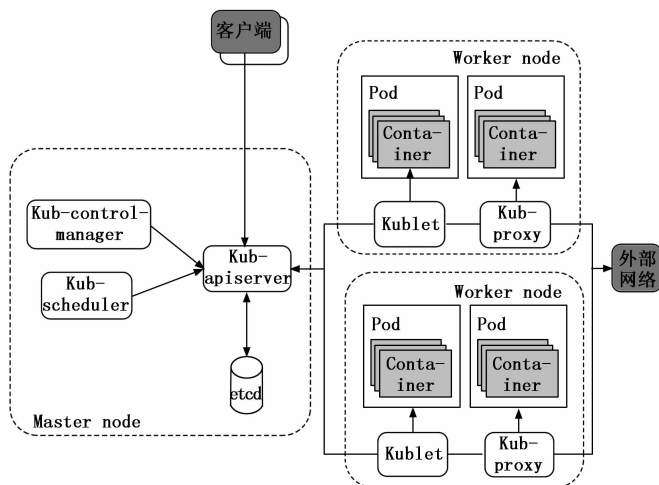


图 3 kubernetes 架构

本文采用了 Kubernetes1.21 版本, 使用了六台物理主机 (centos7.9 操作系统) 构建容器云平台。所有物理主机均需要安装 docker, 以指定 yum 源方式安装 kubelet、kubectl 等组件, 此处使用阿里云的源, 具体硬件配置如表 1 所示。

表 1 容器云平台硬件配置

主机	IP	节点类型	运行服务
master1	10.82.11.50	master 节点	docker、etcd、kube-apiserver、kube-control-manager、kube-schedule、kubectl、kubelet、metrics、calico
master2	10.82.11.51	master 节点	docker、etcd、kube-apiserver、kube-control-manager、kube-schedule、kubectl、kubelet、metrics、calico
master3	10.82.11.52	master 节点	docker、etcd、kube-apiserver、kube-control-manager、kube-schedule、kubectl、kubelet、metrics、calico
worker1	10.82.11.53	worker 节点	docker、kubelet、kube-proxy、calico
worker2	10.82.11.54	worker 节点	docker、kubelet、kube-proxy、calico
worker3	10.82.11.55	worker 节点	docker、kubelet、kube-proxy、calico

## 2.3 高可用集群

应用服务运行在容器之中, 需要容器以极高的可靠性保障业务的连续性。Kubernetes 是容器云平台的核⼼, 控制节点是 Kubernetes 管理功能的关键所在,

其中最关键的是 etcd 键值数据库, 负责保存各节点的关键数据, 用于配置共享和服务发现。针对系统中的关键节点和关键功能, 采用“多控制节点 + etcd 集群”的方式, 来提高整个系统的高可用性和安全性。<sup>[17-18]</sup> Kubernetes 高可用集群架构如图 4 所示。

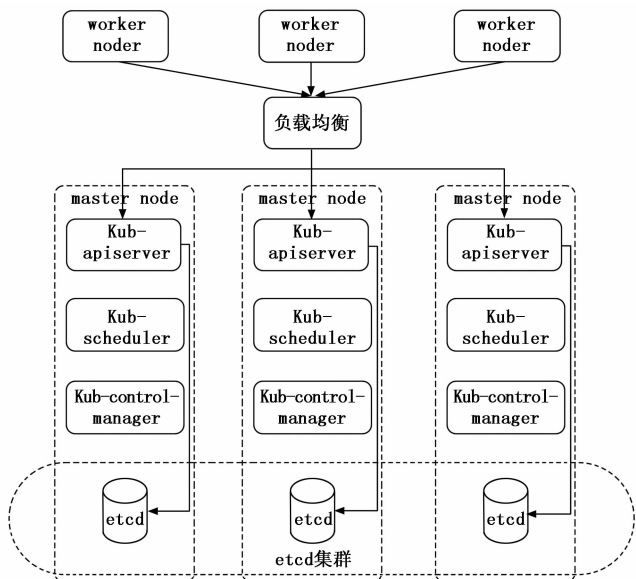


图 4 kubernetes 高可用集群架构

Kubernetes 的工作节点通过内部的负载均衡机制连接到控制节点上, 多控制节点模式下只有所有控制节点同时故障才会影响平台运行。多控制节点的实现是在控制管理器、调度器修改 leader-elect 参数。在多控制节点模式下, 系统会自动选举出主控制节点 (master leader), 如果主控制节点宕机导致系统异常, 则会自动选举出新的主控制节点<sup>[19-20]</sup>。

etcd 集群借鉴了 Hadoop 中 ZooKeeper 的设计理念, 其核心原理是使用 Raft 协议保障节点的一致性<sup>[21]</sup>。etcd 集群中只有 1 个有效的主节点, 主节点负责处理所有来自系统的写操作并对状态机进行维护, 通过 Raft 协议保证主节点对状态机的改动会可靠同步到集群中的其他节点。

## 2.4 外接分布式存储

容器中的数据是非永久保存的, 当容器宕机崩溃时, kubelet 会以镜像的初始状态重新启动容器, 容器中的数据将丢失。为保障容器数据的存储安全, Kubernetes 提供了两套机制: 一是为容器分配临时存储卷, 二是通过容器存储接口 Container Storage Interface (CSI) 提供外部存储服务<sup>[22-23]</sup>。卫星遥感数据总量大需要长期保存, 本文采用第二种方法, 利用分布式存储的核心组件之一 HDFS NFS Gateway,

将 HDFS 存储空间通过 CSI 机制以 NFS 的形式挂载至 Kubernetes, 为容器云平提供外部独立的分布式存储服务。外部存储接口主要依靠 PV 和 PVC 实现, 其中 PV 是底层网络共享存储的抽象, 将共享存储定义为一种供容器消费使用的资源对象, 而 PVC 是对存储资源的申请, 容器可以通过 PVC 的形式申请特定的存储空间和访问模式。

卫星遥感数据单幅数据量大且有其独特的数据格式, 因此在接入容器云平台时需要进行适当改造。以 GF7 号卫星为例, 原始数据以 tar.gz 压缩包形式保存约为 2G, 其中卫星拍摄的影像数据 (jpg 格式) 占单幅数据总量的 90% 以上, 仅在数据处理时才进行读写操作, 而相关的卫星数据参数以 xml 格式保存, 在数据存储、查询、筛选、读写、处理时需要频繁访问。从读取效率、备份安全、存储扩展等方面考虑, 卫星遥感数据不适合直接存储在容器云平台内。因此在采用外接分布式存储的解决方案下, 卫星遥感影像数据经处理后存入独立部署的 HDFS 分布式存储系统, 通过定义 PV、PVC 实现存储系统与容器云平台的对接。

为了提高查询效率, xml 卫星数据参数存入 HDFS 分布式存储系统中的 HBase 数据库中, jpg 卫星影像则直接存入 HDFS 分布式存储系统, 存储路径保存在对应的 HBase 数据库中。HBase 数据库目录直接加载至 Kubernetes 的临时存储卷中供相关容器访问, 需要读写卫星影像数据时, 首先从 HBase 中查询到保存路径, 再调用 HDFS 分布式存储的接口进行操作。定义 PV、PVC 并关联至 Kubernetes 的主要配置命令如下:

```
[root@master ~] mount -f nfs -o ver=3, proto=tcp,nolock,noacl, sync 10.82.8.92:/
[root@master ~] oc describe pv mypv1
[root@master ~] oc get pv mypv1 -o json
[root@master ~] cat newpv.json
[root@master ~] oc creat -f newpv.json
[root@master ~] cat newpvcl.json
[root@master ~] oc creat -f newpvcl.json
[root@master ~] oc get pv
```

## 3 分析展望

本章重点对容器云平台的性能效率进行了简要测试比较, 并进一步结合当前卫星遥感行业现状以及虚拟化平台的应用特性, 分析了容器云平台的应用前景以及面临的实际困难。

### 3.1 容器性能测试

容器相较于虚拟机最大的优势在于轻量化内核、快速化部署。以常用容器官方镜像版本为测试对象, 单一容器测试列表如表 2 所列。

表 2 单一容器测试

镜像名称	启动时间/s
MySQL:5.8	2.8
Tomcat:8.5.24-jre8	2.9
MongoDB:3.6.1	2.7

其中启动时间是指容器由 Penning 状态转变为 Running 状态所需的时间, 测试时容器云平台上没有运行其他非测试容器。根据测试结果可知, 测试容器均能正常启动, 耗时均在 3 秒内启动速度非常快。多容器并行测试选择把表 2 中的镜像均启动 100 个容器, 测试结果如表 3 所列。

表 3 多容器测试

镜像名称	平均时间/s
MySQL:5.8	2.4
Tomcat:8.5.24-jre8	2.8
MongoDB:3.6.1	2.9

通过对比表 2 和表 3 的测试结果可以得知, 随着容器规模的扩大, 容器的启动速度并没有显著下降, 平均启动时间都在 3 秒内, 启动速度远超过虚拟机达到了快速部署的预期, 为高并行应用的开发部署创造了有利条件。

### 3.2 容器功能测试

本项测试这要通过为卫星遥感业务流程来验证容器云平台的功能性, 同时对比相同条件下虚拟机的性能效率。测试环境里 HDFS 分布式存储系统中已导入 260 景 GF7 卫星影像数据作为测试对象, 分别在容器环境和虚拟环境下部署了基于 Web 的卫星影像共享查询系统, 查询系统运行后调用 ElasticSearch 查询位于 HDFS 分布式存储系统中的 GF7 卫星影像数据, 具体测试内容为以下两项:

- 1) 遍历查询整个 GF7 测试数据库中的所有卫星影像数据。
- 2) 查询 GF7 数据库中左上角经度在 (40, 49), 且传感器类型为 mux 的卫星影像数据 (即 rt\_lat 为 (40, 49), 且传感器类型为 sensorid='MUX')。

表 4 业务测试

	容器	虚拟机
查询全部(s)	0.3264	0.4802
查询 rt_lat 在(40,49)且 sensorid='MUX' (s)	24.554	32.067

通过表 4 的测试结果可以看出, 卫星影像共享查询系统部署在容器, 并能够正常调用且访问外接分布式存储系统内的卫星遥感影像数据, 充分证明容器云平台功能正常, 已经可以初步实现对卫星遥感数据的加工处理。对比卫星影像共享查询系统在容器与虚拟机中部署后, 容器中的卫星影像共享查询系统在卫星遥感影像数据查询和读写速度略优于虚拟化平台。

### 3.3 应用前景

#### 1) 微服务:

微服务架构是将单体应用拆分成多个不同功能的微服务模块, 从而降低系统的耦合性、复杂度, 更加便于开发人员部署维护。每个微服务部署到单独容器中, 服务之间可独立开发部署, 支持不同的开发语言, 易于后期维护和功能拓展。目前最常见的微服务案例是 Web 服务, 基于 B/S 架构的卫星遥感应用服务可以优先考虑改造迁移至容器云平台。

#### 2) 深度学习:

深度学习在卫星遥感方向上有着广阔的应用前景, 比如基于影像的地质灾害自动识别、山体滑坡自动监测等。容器可以轻松解决机器训练过程中环境配置、资源分配等一系列问题, 是当前进行深度学习算法研究所用的主流架构。目前容器已支持 TensorFlow、Caffe、Caffe2、PyTorch 等主流深度学习的训练框架, 能够为卫星遥感数据的深度学习研究提供平台与技术支持。

#### 3) 弹性伸缩:

对于一次性执行运算任务, 传统模式单一任务占据整个服务器, 造成了极大资源浪费, 容器快速创建、灵活销毁的特点更适用于这种按需使用的场景。需要运算时创建容器执行任务, 运算完成后容器退出释放其占有的硬件资源。

对于大规模并行计算任务, 容器云平台能够根据资源的负载情况进行动态调节, 避免了流量激增扩容不及时资源耗宕机, 也避免了平时大量闲置资源浪费, 提高了硬件资源利用率。

#### 4) 高效运维:

通过镜像管理运维人员可以轻松地对平台内的容器及部署在容器内的业务进行部署管理。需要升级容器内的业务时, 运维人员只需选择相应镜像版本和副本数量, 即可实现自动升级且业务平滑不中断, 升级后若有问题还可选择回滚至旧版本。

### 3.4 面临的问题

#### 1) 速率瓶颈:

虚拟化与容器提升了物理主机的硬件资源利用率, 代价则是更多的数据交互与网络开销。当业务高峰期并行数量大时, 物理主机用来支撑容器的服务资源就会紧张, 造成队列堵塞, 前端用户的感受就是应用卡顿。由于卫星遥感单幅影像数据量大, 总数据量可达 PB 级, 不可避免需要采取独立部署的分布式或集中存储系统, 单独管理数据存储。无论是虚拟化平台还是容器云平台, 都是通过内部网络进行访问读写操作, 网络带宽与数据吞吐速率始终是制约卫星遥感数据处理能力上限的瓶颈。

#### 2) 应用容器化:

容器的优势在于轻量化的内核、便捷的封装部署, 但并不是所有的应用服务都适合或者能够被容器化, 比如说 IO 交互密集的数据库、某些 license 绑定固定硬件的软件。此外, 将应用服务迁移至容器部署需要对代码进行容器化改造, 也需要人力和时间成本付出, 比如地质行业最常用的专业软件 ArgGIS、GXL 等, 短期内都无法实现容器化部署, 这也制约了容器技术在整个行业中的推广应用。

#### 3) 隔离安全:

容器由于共享操作系统内核及驱动, 容器内某应用服务的非法运行或者故障报错, 可能会导致物理主机的操作系统异常, 进而影响到其他容器及容器内的应用服务。此外, 也更容易被外部攻击通过安全漏洞攻击操作系统, 进而影响上层应用服务。因而, 在当前的应用场景下, 容器还不太适合公有云的运行环境, 如有外部接入还需配套设计严格的安全防护机制。

## 4 结束语

基于 Kubernetes 的容器云平台是容器技术在卫星遥感行业的应用探索, 已能够初步实现对卫星遥感数据的处理加工, 提高了物理主机的硬件资源利用率, 降低了后期运维管理成本, 并在微服务、深度学习等方向上展现出现了巨大潜力。随着容器技术的进一步发展普及, 其所面临的问题和对应的解决方案也

会越来越多。基于 Kubernetes 的容器云平台为进一步建设功能更为丰富、实用性更强的容器云平台打下了良好的基础, 对推进卫星遥感行业后续传统业务上云改造具有较大借鉴意义。

#### 参考文献:

- [1] 金九平, 李婉嘉, 戴 橙, 等. 基于 kubernetes 的海洋遥感数据产品服务平台设计与实现 [J]. 海洋信息, 2020, 35 (1): 27-34.
- [2] 钟伟宏, 孙甲琦, 朱宏涛, 等. 基于 Kubernetes 的航天地面应用软件架构设计与实现 [J]. 遥测遥控, 2021, 42 (3): 46-54.
- [3] 兰智博. 基于 Kubernetes 的容器云平台的设计与实现 [D]. 北京: 北京邮电大学, 2020.
- [4] BERNSTEIN D. Containers and Cloud: From LXC to Docker to Kubernetes [J]. IEEE Cloud Computing, 2014, 1 (3): 81-84.
- [5] PAHL C. Containerization and the PaaS Cloud [J]. IEEE Cloud Computing, 2015, 2 (3): 24-31.
- [6] 杨保华. Docker 技术入门与实战 [M]. 北京: 机械工业出版社, 2018: 3-6.
- [7] JOY A M. Performance comparison between Linux containers and virtual machines [C] //2015 IEEE International Conference on Advances in Computer Engineering and Applications, 2015: 342-346.
- [8] 陈金光. 基于阿里云的 Kubernetes 容器云平台设计与实现 [D]. 杭州: 浙江大学, 2018.
- [9] 翁涅元, 单杏花, 阎志远, 等. 基于 Kubernetes 的容器云平台设计与实践 [J]. 铁路计算机应用, 2019, 28 (12): 48-53.
- [10] 于泽萍. 面向微服务架构的容器云平台设计与实现 [D]. 哈尔滨: 哈尔滨工业大学, 2018.
- [11] 章仕锋. 基于微服务架构的分布式档案系统设计与实现 [D]. 宁波: 宁波大学, 2019.
- [12] 王骏翔, 郭 磊. 基于 Kubernetes 和 Docker 技术的企业级容器云平台解决方案 [J]. 上海船舶运输科学研究所学报, 2018 (9): 51-57.
- [13] 辛园园, 钮 俊, 谢志军, 等. 微服务体系结构实现框架综述 [J]. 计算机工程与应用, 2018, 54 (19): 10-17.
- [14] 谢超群. 基于 Kubernetes 容器云的应用研究 [J]. 西安文理学院学报 (自然科学版), 2020, 23 (4): 37-40.
- [15] 龚 正. Kubernetes 权威指南 [M]. 北京: 电子工业出版社, 2017: 12-18.
- [16] 何震苇, 黄丹池, 严丽云, 等. 基于 Kubernetes 的融合云原生基础设施方案与关键技术 [J]. 电信科学, 2020, 36 (12): 77-88.

(下转第 220 页)