创新开源企业容器云平台研究应用

贺　英[1],陈　航[2]

(1.湖南广电网络云数据有限公司,湖南长沙 410005;
2.湖南邮电职业技术学院,湖南长沙 410015)

【摘要】基于Kubernetes这款开放源码的容器技术,首先设计了一种容器云计算平台的体系结构,包含OS操作系统、云管服务、应用平台、数据平台、分析平台五大功能模块,并抽象了用户的平台信息,从而达到了组件间拉通的目的,其次剖析了应用程序的全过程管理、实体计算机的资源管理、多层租户用户的管理等三方面关键技术,最后提出了企业级App开发、数据挖掘分析应用和企业私有云中的应用等主要应用场景。实施结果表明,容器云平台具有较高的商业应用价值,可助力企业的数字化转型升级。

【关键词】计算多租户;容器云;OLTP/OLAP;中间件服务
【doi:10.3969/j.issn.2095-7661.2022.03.011】
【中图分类号】TP393.09　　【文献标识码】A　　【文章编号】2095-7661(2022)03-0043-04

Research and Application of Innovative Open Source Enterprise Container Cloud Platform

HE Ying[1], CHEN Hang[2]

(1. Hunan Radio and TV Network Cloud Data Co., Ltd., Changsha, Hunan, China 410005;
2.Hunan Post and Telecommunication College, Changsha, Hunan, China 410015)

Abstract: Based on kubernetes, an open source container technology, this paper first designs the architecture of a container cloud computing platform, including five functional modules: OS operating system, cloud management service, application platform, data platform and analysis platform, and abstracts the user's platform information, so as to achieve the purpose of communication between components. Secondly, it analyzes three key technologies: the whole process management of applications, resource management of entity computers and the management of multi-layer tenant users. Finally, the main application scenarios, such as enterprise level app development, data mining analysis application and application in enterprise private cloud, are proposed. The implementation results show that the container cloud platform has high commercial application value, which is conducive to and helps enterprises' digital transformation and upgrading.

Keywords: calculating multi-tenancy; container cloud; OLTP/OLAP; middleware services

1 研究基础

在云基础平台层IAAS方面,经典的OpenStack等开源云平台往往以完整虚拟主机、完整虚拟操作系统资源等形式进行编排和部署,属于重量级的IT基础设施,资源浪费较严重、可移植性较低[1]。云技术不断发展,目前新一代轻量级Kubernetes等开放源码的容器技术已经被大量用于DevOps和微服务中。经典云平台所支撑的OLTP事务型服务、OLAP在线分析型服务、机器学习等需求都需要在轻量级平台内进一步解决和优化。以前的业务需要集中在物理集群X86服务器或独立云主机上,导致了业务的孤岛化和资源使用不均衡问题,也需在轻量级平台内重新考虑和解决。将企业关键主要的业务服务和分析服务整合到Kubernetes

[收稿日期] 2022-05-08
[作者简介] 贺英(1979—),男,湖南长沙人,湖南广电网络云数据有限公司项目总监,硕士,研究方向:大数据算法、金融风险控制。
[基金项目] 2021年湖南省高校思想政治工作精品项目"基于'易班'平台的学生管理信息化模式构建"(项目编号:21JP076)。

(Docker底层)容器内,利用其标准化、集约化、自动化的“集装箱”特性形成一个灵活的容器云层,是许多公司IT基础架构的一个新技术选项[2]。

优化设计的4个主要原则:Kubernetes APIs是声明性的而非命令性的;Kubernetes控制平面是透明的,没有隐藏的内部API;满足用户的需求;可移植的工作负载。

2 容器云计算平台体系结构设计

容器云计算平台体系结构如图1所示,容器式企业云计算的整体结构分为五大块:

1)OS操作系统:基于Linux等实体服务器操作系统,针对Kubernetes进行了自定义二次优化,并添加了相应的网络、存储、安全和监控等方面的支持,为Kubernetes在企业级应用的整体架构奠定了基础。

2)云管服务:将底层操作系统的所有功能都进行抽象化包装,将复杂度隐藏起来,并生成一个简单的功能性界面;提供信息安全、系统维护、运行监控、账务处理、容灾处理等功能。

3)软件(应用平台)业务:应用平台服务为开发人员提供通用的、共性的软件开发和检测工具,并通过角色或单点授权帮助开发者向软件市场中的用户发布成果软件。

4)数据(数据平台)业务:数据平台服务整合了通用的OLTP/OLAP业务,使客户能够迅速建立起主要的数据库业务,包括大宽表、数据集市、非结构化检索、小型数据仓库、实时流计算、开发套件、图数据库、数据资产目录等。

5)解析(分析平台)业务:分析平台服务整合了常见的报表工具、机器学习组件及入口统一门户信息,便于使用者操作和存取资料,形成经营报表、预测模型等成果。

通过上述五大功能的协作,可以在公司内部建立更高层次的App系统,例如财务系统、客户关系管理系统、办公系统、电子商务系统等。

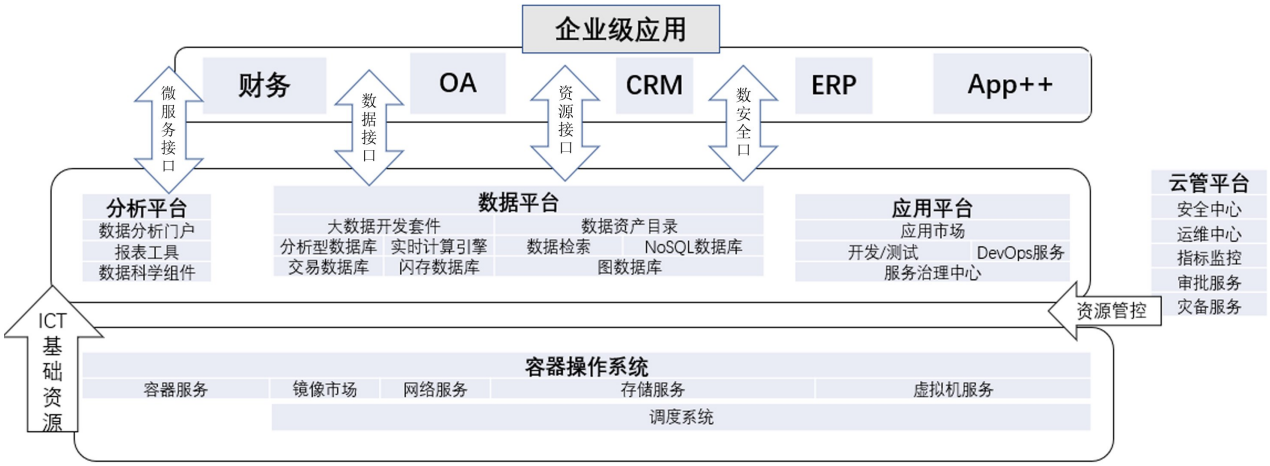

图1 容器云计算平台体系结构图

3 关键技术

Kubernetes已建立基础的抽象建模和相应的函数,根据客户共性业务的定制经验,文章从管理应用、调度系统和多用户间的分离等几个方面进行了阐述[3]。

3.1 应用程序的全过程管理

如果要在一台服务器上运行多个任务,传统的方法是将其划分为多个虚拟机,使用每个虚拟机来运行一个任务。但是虚拟机启动很慢,因为它们必须启动整个操作系统,这要花费数分钟的时间,而且这会占用大量资源,因为每个虚拟机都需要运行一个完整的操作系统实例。容器则提供了某种类似操作系统的行为,但是速度更快,因为启动一个容器就是启动一个系统进程。

在Kubernetes中可以像搭积木一样进行应用资源的编排,具体以Pod和Stack两种方式实现:

1)Pod编排。适用于紧耦合的服务组,保证一组服务始终部署在同一节点,并可以共享网络空间和存储卷。同一个Pod内的容器可以通过localhost访问彼此服务,共享网络空间,容器的端口不互相冲突;对于同一个存储卷,可以被同一个Pod的多个容器操作。通过Pod编排,不需要重新构建镜像,就可以把多个服务进行整合;如果一个容器推荐仅包含一个进程,那么Pod更像是可以容纳多个进程的虚拟机。

2)Stack编排。设计上与Docker Compose相似,但可以支持跨物理节点的服务之间通过API进行网络通信。

以上两种编排均支持用YAML文件描述多个容器及其之间的关系，定制各个容器的属性，并可联动部署及运行。

另外Kubernetes采用了声明API的方式，用户可以将声明的档案交给Kubernetes来建立一个程序，且需要多个声明档案来支撑该程序的运作。技术人员在应用程序部署期间，需提交一个包含自定义镜像档案的配置方式。在App运行后，使用者可以通过组态中心进行系统的修改。传统的维护模式是使用SSH命令行工具登陆主机，而容器云系统作为一个稳定的基础架构，对安全性有很高的需求，通过更改服务的配置来实现对服务器App的调整。

3.2 实体计算机的资源管理

3.2.1 Resources估计程序

Kubernetes采用Request/Limit来限定CPU和内存，Request表示请求计划程序所保留的资源量，Limit表示应用程序的最大资源使用率，但Request和Limit的估计差别会引起稳定性问题，例如内存溢出等[4]。由于此问题，本文提出了一个利用历史时期的平均与最大的资源估计方法，进行了估计计算，从而解决资源浪费、资源溢出等问题。在系统本身和重要业务方面，可以通过添加业务的特性，在监测到的数据量超出一定范围时发出报警，并引入人工运维处理，但是不会将系统本身业务和重要业务自动剔除。

3.2.2 灵活储存计划

平台支持识别不同的存储资源，包括HDD和高速SSD、闪存等，将不同类型的存储资源分级。不同级别的存储资源能够分别管理，同一租户可以申请不同级别的存储资源；不同级别的存储资源能够被分别申请，提供不同的逻辑卷配额；每个容器进程有各自的存储空间，相互独立，无法直接相互访问；支持Pod存储空间持久化，容器销毁时，数据仍然能够保留，并能够进行数据恢复；各容器进程的存储空间能够限制大小，保证容器进程使用的最大空间不超过配额上限。存储卷管理支持创建存储卷信息、删除存储卷管理信息、修改存储卷信息、显示存储卷信息列表，包括名称、类型等。

1）本地存储。平台能够对集群各个服务器的本地存储资源进行管理，并按租户进行分配。具体包括：能够管理服务器的本地存储资源，包括SATA、SAS、SSD等；能够将本地存储按需分配给不同的租户使用；支持Convoy本地存储框架；支持自定义本地存储类型，如根据需求指定RAID等级、不同应用间进行物理隔离等；支持对容器存储空间的动态存储供应；支持软件RAID和硬件RAID；支持快照功能，可以对持久化数据卷进行备份和还原。

2）分级存储。因为集群的硬件复杂性，可能同时具有SATA、SAS、SSD、NAS等各类存储资源，各类存储资源价格不同，性能不同，空间大小也不同，因此平台支持对存储资源的分级管理功能。具体包括：能够识别不同的存储资源，将不同类型的存储资源分级；不同级别的存储资源能够分别管理，一个租户能够申请不同级别的存储资源。

3）分布式文件系统支持。平台支持常见的分布式存储系统，包括以下功能：支持对象存储，如视频、图片数据存储；支持文件存储，文件共享；支持数据备份机制；支持块存储，可以作为无HA服务的数据卷，保存应用的配置；支持多副本数据存储方案，保证数据的高可用。

4）存储隔离。平台提供对各个租户间的存储隔离，支持各个租户能够独立使用各自的存储资源；运行时保证存储IO/IOPS的隔离性，保证各个应用的运行稳定性。具体包括：存储空间隔离，平台支持各个租户有独立的存储空间，相互隔离拒接直接访问；各个租户的存储空间能够限制大小，保证租户使用的最大空间不超过配额上限；不同级别的存储资源能够分别申请，提供不同的逻辑卷配额。读写IO隔离，平台支持各个服务应用的读写IO隔离，保证各个租户运行的应用相互独立，不受其他应用IO争夺的影响。

3.2.3 灵活配置资源

灵活配置的系统可以有效地提升系统的使用效率和满足系统的突发接入要求。具体包括：

1）资源的固化：最基础的资源分配，为保障基本运行提供的服务。

2）灵活的资源：在满足客户需求的情况下，灵活地扩展容量，在发生资源碰撞的时候，利用不同服务的不同优先权来处理冲突。

3.2.4 运行维护监控

平台采用开源Prometheus作为监控软件，主要在以下各层面实现监控，一旦指标突破某个阈值，将能够按照配置进行告警信息的发送[5]：

1）基础设施层：监控各个主机服务器资源（包括Kubernetes的Node和非Kubernetes的Node），如CPU、内存、网络吞吐和带宽占用，磁盘I/O和磁盘使用等指标。

2）中间件层：监控独立部署于Kubernetes集群

之外的中间件，例如：MySQL、Redis、RabbitMQ、Search、Nginx等。

3）Kubernetes集群：监控Kubernetes集群上部署的应用。监控指标主要分为两类，一类属于性能指标，包括容器相关的性能指标数据、Pod相关的性能指标数据、主机节点相关的性能指标数据；二类属于状态指标，包括服务健康状态监控、Deployment相关的健康状态、Pod的健康状态、主机Node节点的健康状态等。

3.2.5 负载均衡管理

平台设置负载均衡管理模块，支持负载均衡列表查看、添加负载均衡、停止负载均衡、删除负载均衡、编辑负载均衡等功能。

3.3 多层租户用户的管理

非定制开源的Kubernetes自身不存在“租户”的概念，仅对“命名空间”的资源授权进行了管理，因此不能直接满足多层复合的租户体系结构[6]。Kubernetes中也没有设定使用者的概念，使用服务账号来进行系统的调度。数据事务服务、数据分析服务等都要有租户和使用者的角色，而且应用端App还要求Auth、LDAP、Access Token等身份验证，基于此，设计并提供6种功能：

1）租户：属于基础单元，从容器中获得一个资源，一个单独的账号。

2）项目：属于操作单元，包括计算、存储、安全、网络等必须资源的组合，以及对资源使用的付费记录。

3）使用者：属于机构的成员，也是一个系统成员，由平台管理者加入，是容器云的直接操作者，同时可以共享一个租户的总限额。

4）使用者群：多名使用者的逻辑聚类，可以是租户内的成员，也可以是租户内的项目。

5）角色：一系列授权的集合，以便于管理，如系统管理员、开发管理员、程序员、普通用户等。

6）管理操作员：管理操作员主要是为建立、维护和管理容器平台，通常是为使用者增加身份验证和权限，并与容器平台的各个层次的业务进行连接。

4 主要应用场景

4.1 应用程序开发的平台

平台包含了DevOps和MicroServices的发展，能够实现迭代升级和快速升级，并且在系统组织或团队确认后，这些服务可以被直接使用。在已存在的产品（现存系统）或新应用程序的部署运行中，针对在软件市场或应用商店上发布的App进行安全扫描等，以保证软件的安全性和遵从性。用户也可以根据具体业务的需要进行相应的软件定制，定制过程中可直接使用平台内各项现有资源，而无需单独采购或引入。

4.2 内部能力开放

数据资源、数据资产丰富的公司可以通过平台能力建立并运营数据业务云，为第三方需求者提供数据挖掘与分析服务赋能，包括公共数据、行业分析报告、公共共性模型等业务。第三方需求者以租客的形式租用信息、资料、算力、存储空间等，并根据实际使用情况进行账单计费及结账。

4.3 生态开放合作

该平台具有多租户模型、安全策略、计费和支付机制，能够向上下游的公司和机构提供计算资源、数据、模型、开发环境以及工具软件的支持。行业生态内相关的企业单位可以结合自有的数据资源、自有的算法模型软件，拉通本平台内的各项资源，采用联邦（机器）学习的方式进一步优化已有的算法模型，提升模型准确率和生产力。

5 结语

Kubernetes是一种创新开源的、可定制的、稳态的IT基本技术与数字化基础设施框架，但在实际应用中需要由掌握容器化、云原生、微服务等工程技术的工程师团队来设计、部署、实施、维护和改进，存在一定的学习成本和技术门槛，目前主要在互联网企业使用较多。随着“互联网+产业”的不断推广与技术赋能，Kubernetes方式的容器云逐步为传统企业所接受。

【参考文献】

[1]刘国成,吴丹.基于OpenStack的中小企业服务云平台架构研究[J].吉林大学学报(信息科学版),2020(6):709-713.

[2]张有帅,佘葭,尹雪龙.基于Kubernetes的容器云平台研究与设计[J].电子设计工程,2021(22):180-183,188.

[3]李华东,张学亮,王晓磊,刘惠,王鹏程,杜军朝.Kubernetes集群中多节点合作博弈负载均衡策略[J].西安电子科技大学学报,2021(6):16-22,122.

[4]张世杰.基于Kubernetes的数据库集群容器化技术研究[J].自动化应用,2021(7):64-66.

[5]郭建磊,车学董,王宁宁.基于Prometheus的工业云平台监控告警服务系统[J].信息技术与信息化,2021(12):142-145.

[6]梁进科,陈路路,王一,张建廷.容器云环境下可视化编排技术[J].计算机与网络,2021(23):58-60.