

云原生在物联网边缘计算中的应用

郑响萍, 蔡海军

(浙江理工大学科技与艺术学院, 浙江 绍兴 312369)

✉xiangping2002@163.com; china_cai@163.com



摘要: 随着智能终端的大量普及, 大数据的传输、计算和数据安全问题的解决方案是当前的研究热点之一。基于Spring Cloud云原生微服务和容器化技术, 提出一套容器化云原生边缘系统解决方案。可就近计算局部产生的海量数据, 从而减少数据的传输和中心节点的计算量, 解决了中心化架构的大数据传输、计算和安全问题。该方案可以在电力、冷链物联网系统等场景下使用。

关键词: 物联网; 智能终端; 边缘计算; 云原生

中图分类号: TP311 **文献标识码:** A

Application of Cloud Native in Edge Computing of Internet of Things

ZHENG Xiangping, CAI Haijun

(Keyi College of Zhejiang Sci-Tech University, Shaoxing 312369, China)

✉xiangping2002@163.com; china_cai@163.com

Abstract: With the popularization of intelligent terminals, the solution of big data transmission, calculation and data security is one of the current research focuses. Based on Spring Cloud cloud native micro-services and containerization technology, this paper proposes a set of containerized cloud native edge system solutions, which can compute locally generated massive data nearby, thus reducing the amount of data transmission and central node computing, and solving the large data transmission, computing and security problems of the centralized architecture. The proposed solutions can be used in scenarios such as power and cold chain IoT (Internet of Things) systems.

Keywords: Internet of Things; intelligent terminal; edge computing; cloud native

1 引言(Introduction)

物联网技术的核心是网络通信和感知技术, 实现人、机、物的泛在连接, 提供信息感知、数据传输、计算处理等服务^[1]。物联网起源于20世纪90年代末, 源自美国麻省理工学院, 早期以物流系统为背景, 把所有物品通过射频识别等信息传感设备与互联网连接起来, 实现智能化识别和管理^[2]。在2005国际电信联盟(International Telecommunication Union, ITU)发表的ITU Internet Reports 2005: The Internet of Things正式确立了物联网概念, 报告提出信息通讯将发展到实现人与物、物与物之间的连接, 无所不在的物联网通信时代即将来临^[3]。

随着计算机软硬件和网络通信技术的发展, 智能化设备已得到普及, 现有中心化系统架构已无法支撑快速产生的海量数据的传输、存储和计算, 而敏感数据因安全问题也不便传输到中心节点。本文提出容器化云原生边缘计算解决方案, 通过边缘系统就近计算, 减少数据传输的网络延迟, 解决数据安全问题, 降低中心服务器计算和存储数据量压力。以冷链系统为例, 实现快速响应、计算强大、弹性智能的云原生边缘计算平台。

2 云原生边缘计算方案介绍(Introduction to cloud native edge computing solutions)

云原生在2013年由Pivotal公司的Matt Stine首次提出,

用以区分为云而设计的应用和云上部署传统应用，随后于2015年7月，云原生计算基金会(Cloud Native Computing Foundation, CNCF)成立，致力于云原生应用推广和普及。CNCF对于原生云的定义为“云原生技术有利于各组织在公有云、私有云和混合云等新型动态环境中，构建和运行弹性扩展的应用。云原生的代表技术包括容器、服务网格、微服务、不可变基础设施和声明式API(Application Programming Interface)。这些技术能够构建容错性好、易于管理和便于观察的松耦合系统。结合可靠的自动化手段，云原生技术使工程师能够轻松地对系统做出频繁和可预测的重大变更”。

云原生的核心是微服务、服务网格和容器化技术。业界有不少微服务的实现方案，Spring Cloud是常用的微服务解决方案，其提供了微服务工具集合，如服务注册、服务发现、负载均衡、API网关等，将微服务通过Docker容器化部署，通过DevOps系列工具，持续交付部署。其整体架构如图1所示。

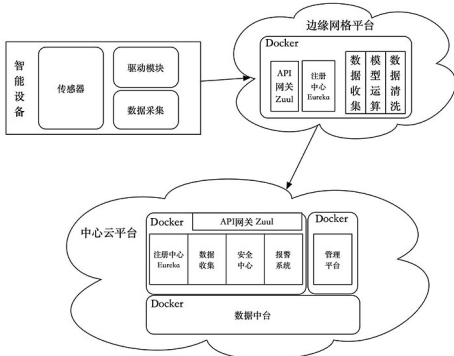


图1 云原生边缘计算架构图

Fig.1 Architecture diagram of cloud native edge computing

2.1 微服务

系统采用Spring Cloud微服务架构，Spring Cloud是一套完整的微服务解决方案。基于Spring Boot框架实现了微服务容器，包含了一系列简单易用的微服务组件，例如服务治理、服务网关、配置中心、负载均衡、服务跟踪等^[4]，该系列组件经过Netflix等公司多年实践经验总结。

2.1.1 注册中心

微服务架构的服务提供者，通过Spring Cloud Eureka实现服务注册和发现。Eureka是Netflix中的一个开源框架，主要提供服务注册、服务发现和状态同步三大主要功能。

服务提供者在服务启动时，会通过Eureka Client向Eureka Server注册服务信息，Eureka Server会保存该服务提供者的服务信息，为后续服务消费者提供查找服务。服务消费者在服务启动时，会通过Eureka Client向Eureka Server获取服务注册表，并缓存到本地，默认情况下每30秒更新一次注册表，以便消费者从本地获取服务地址，直接调用服务提供者的服务。这种方式可以有效支撑分布式节点的弹性扩容和缩容，服务提供者可根据服务能力的需求调配数量，对于物联网应用来说，终端数量的增减非常频繁，弹性扩容能力

尤为重要，如图2所示。

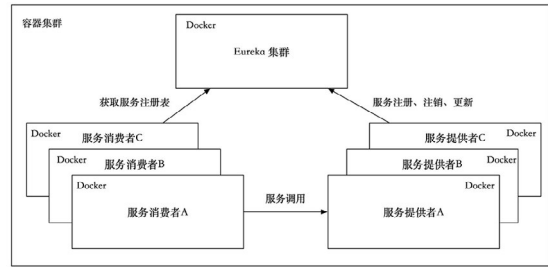


图2 服务注册和发现

Fig.2 Service registration and discovery

2.1.2 负载均衡

系统服务提供者均为分布式微服务部署，多节点服务通过Spring Cloud Ribbon实现负载均衡。Spring Cloud Ribbon是一个基于超文本传输协议(Hyper Text Transfer Protocol, HTTP)和传输控制协议(Transmission Control Protocol, TCP)的客户端负载均衡工具，它基于Netflix Ribbon实现。服务消费者从Eureka获取到服务注册表后，服务消费者获得多个服务提供者的服务地址，消费者在发起服务器请求时，Ribbon提供负载均衡能力，将请求分摊到不同的服务提供者，如图3所示。

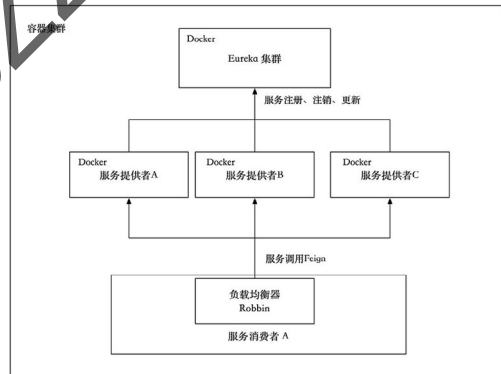


图3 负载均衡

Fig.3 Load balancing

Ribbon提供的负载均衡算法有多种，包括轮询、随机、最小并发量和最小活跃数等，同时还支持自定义算法，在非异构服务器节点场景下，最常用的是轮询算法。考虑到物联网存在多云和异构服务器的情况，需要一种优化算法，而基于动态反馈的负载均衡算法更适合这种场景下使用。根据服务处理反馈，利用滑动窗口算法，动态调配负载任务，有利于集群服务的稳定和性能最大化^[5]。

2.1.3 服务消费

服务的调用基于HTTP协议，Spring Cloud Feign封装了HTTP调用服务，Feign是基于Netflix Feign实现，整合了Spring Cloud Ribbon和Spring Cloud Hystrix。

Feign提供了快捷、优雅的方式调用服务，就像调用本地方法一样。通过整合Ribbon负载均衡和Hystrix延迟与容错服务，让微服务集群能够均衡提供服务，且当服务出现单点问

题时，不至于产生整体服务雪崩，避免级联故障，有效提升分布式系统的稳定性。

物联网边缘计算服务下，集群环境复杂、集群规模庞大，需要整合多云环境，服务易出现单节点不可用或宕机，当不采取容错措施时，易产生因单节点故障而引起整个集群不可用现象，如A服务调用B服务，当B服务不可用时，A服务也将不可用，这种级联现象称为雪崩效应，级联层次越多，容错需求就越大。而Feign在提供HTTP调用服务外，还通过Hystrix提供降级和熔断，服务B出现超时或不可用现象，可快速返回错误信息，通过FallbackFactory等方法，将服务降级为备选方法，保证集群的整体稳定。

2.1.4 服务网关

该系统集群环境间是隔离的，集群间服务通过统一的服务网关对外提供服务能力，服务网关为Spring Cloud Zuul，Zuul是Netflix开源的API网关服务，提供动态路由、监控、弹性、安全等边缘服务框架。网关Zuul作为整个集群服务的唯一出口，增加了接口鉴权服务，只有带有授权信息的消费者才可调用，为整个集群增加了一道防护，如图4所示。

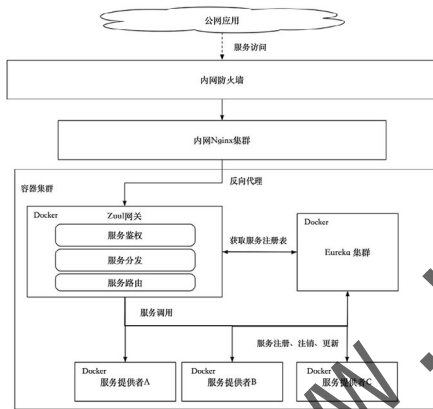


图4 服务网关

Fig.4 Services gateway

2.2 容器化

本方案通过Spring Cloud微服务化后，服务数量增加至十几个，物联网系统服务相对较为复杂，涉及多云环境部署。而随着业务的发展，应用集群服务节点数可达上百个，传统部署的成本非常高，一方面单服务器多进程资源利用率低；另一方面，多云异构环境下，服务器配置、操作系统等环境都不尽相同。运维人员需要掌握每一台服务器的特性，去针对性地优化和配置，运维难度和人力消耗极大。同时随着服务高峰和低谷的波动，服务器的数量需求是弹性的，动态调整服务数量也是一个极大的难题。

综上所述，该系统采用Docker容器化技术部署方案。容器化是一种虚拟化技术，用来隔离运行在主机上不同的进程，使进程之间、进程和宿主操作系统相互隔离、互不影响。Docker将异构的服务器环境部署为统一虚拟环境，将应用创建为镜像服务，部署到任一虚拟环境下^[6]。利用

Kubernetes等容器编排工具，可实现快速扩容和缩容，提高集群弹性能力^[7]。

2.3 DevOps

该系统包含十几个微服务应用，场景下应用数量将更多，容器化可简化部署的复杂度，但部署的数量级依旧很大，DevOps可有效解决以上问题。DevOps是指通过工具协助，快速高效地完成应用的生命周期管理，实现应用系统快速交付和部署。DevOps工具很多，本方案采用Jenkins与Kubernetes相结合，使用Docker容器技术，通过Jenkins持续集成工具编译和构建镜像，并通过Kubernetes容器化编排工具，部署到Docker容器化集群节点中，实现服务节点的监控、快速部署、扩容和缩容^[8]。

2.4 边缘计算

本方案将传统单中心系统架构优化为云原生微服务边缘计算架构。边缘计算架构是指在需求所在地或数据源头的附近，提供最近端的服务，把原本中心节点的应用计算服务，移至网络的逻辑边缘节点来处理，是一种分散式运算架构。将大型中心节点，拆分成更小、更易管理、处理更快的就近计算节点。就近边缘计算节点能有效解决中心节点远程传输带来的网络延迟、数据安全等问题，在靠近数据源头处提供智能分析计算，减少时延，提升效率，提高安全隐私保护。边缘计算获取海量数据后，快速完成数据清洗和脱敏，将少量有效数据传输给中心节点^[9]。

3 系统实现(System implementation)

本方案以冷链系统为例，包括冷链车载系统、冷链仓储系统和冷链管理中心系统三部分。由于生鲜、医疗用品等物品对温度极其敏感，在路途和仓储中需要严格监控温湿度、运输管理人员健康情况等指标。系统须在每台运输设备和每个冷链仓库安装多个温湿度传感器等智能设备，将温湿度、人员健康信息、运输车辆和仓储数据，实时传输到本地系统进行计算和预警，并将汇总数据传输至冷链管理中心系统，其整体系统设计如图5所示。

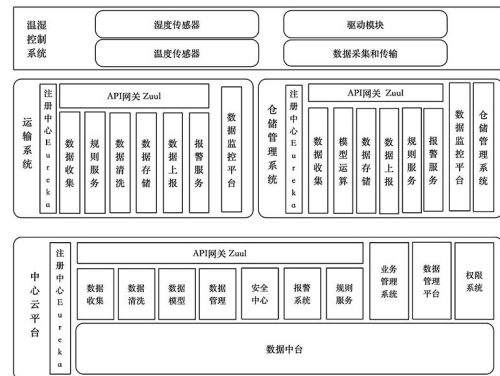


图5 系统介绍

Fig.5 System introduction

3.1 边缘系统

边缘系统包括冷链运输车冷链仓库温湿度监控系统。

每一台冷链运输车部署数个智能温控设备和一台主机，智能设备采集运输物品温湿度和运输人员健康信息，将数据传输到车载主机部署的温控系统，温控系统实时计算获取到的温度和湿度数据，保证温度湿度在合理范围内。如运输物品温湿度或运输人员健康参数出现不达标，马上预警给运输人员和中心云平台进行调控。温度和湿度的规则由中心云平台设定并投放到车载系统，因此冷链运输车的温湿度采集频率可以提升毫秒级，实现实时监控。将温湿度统计到中心云平台设定的周期数据后再统一上报，可减小中心云平台的数据压力。

冷链运输车车载温控边缘系统包括数据收集、规则服务、数据上报、报警服务和数据管理存储等服务。系统为Spring Cloud微服务化架构，服务间调用通过Feign实现，为便于远程快速更新升级，服务通过Docker容器化部署，如图6所示。

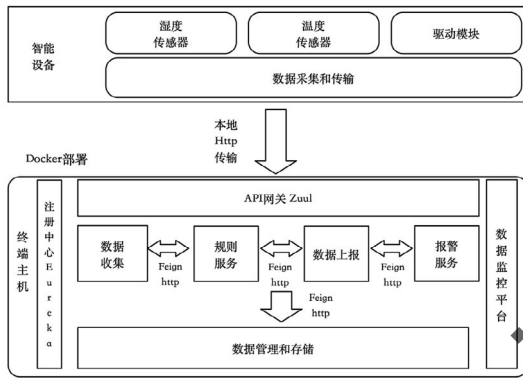


图6 车载温控系统

Fig.6 Vehicle temperature control system

仓库部署温湿控边缘系统包括智能设备、数据采集、温控管理平台、仓库管理系统、物流管理系统等。智能温控设备上传的数据，同样在规则服务中完成风控预警计算，而采集到的数据样本将在本地完成数据清洗和整理，汇总后上报给中心云平台。

服务为Spring Cloud微服务架构，根据仓库的大小，安装一定数量的智能设备。随着智能温控设备安装数量的增加，数据收集服务和规则服务等需要部署多个节点，通过容器化集群部署，只需要增加镜像进程即可实现服务节点的增加，如图7所示。

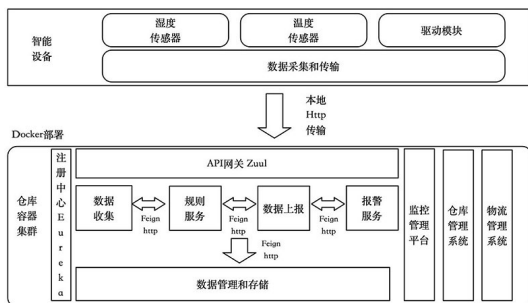


图7 仓库智能管理系统

Fig.7 Warehouse intelligent management system

3.2 中心化平台

中心化平台包括数据收集、数据运算和存储、数据中台、业务管理平台等。边缘计算系统减少了中心化平台采集的数据，将采集数据周期从毫秒级数据降低到分钟级，整体数据运算量减少数倍，但边缘系统数量多，整体采集的数据依然巨大。数据清洗和计算节点数量随着运输车辆和仓库数量的变化而不断增加，高峰期与低谷期数量可能会有数倍之多。为使中心化云平台能具有高弹性计算能力，平台采用Kubernetes作为容器化编排工具，通过其Replication Controller对Pod副本数的控制能力，可以快速增缩节点数，从而弹性地增加和减少集群的计算能力。中心化平台容器集群如图8所示。

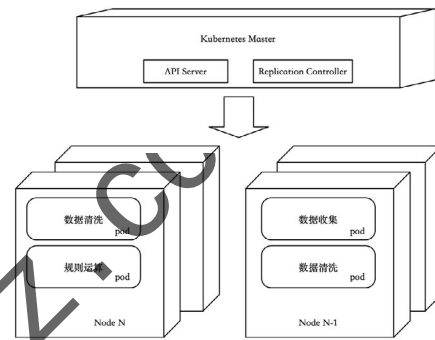


图8 中心化平台容器集群

Fig.8 Centralized platform container cluster

中心化云平台通过Zuul网关提供数据收集等HTTP接口服务，通过Netflix设定降级和熔断，避免雪崩效应。

4 应用部署(Application deployment)

本方案采用阿里云服务器ECS部署，采用一台2核4 GB服务器、四台2核8 GB云服务器和一台MySQL关系型数据库；在2核4 GB服务器上搭建Jenkins环境、Harbor、Maven仓库和GitLab等管理工具；在四台2核8 GB服务器上搭建Kubernetes集群环境和容器化环境。

微服务包括冷链车载系统、冷链仓储系统和冷链管理中心系统三大系统。构建Eureka代码上传到GitLab，通过Jenkins拉取Git代码并打包成Docker镜像上传到Harbor，最后Jenkins执行Kubernetes插件，使Kubernetes拉取镜像创建服务。构建集群Eureka双节点Eureka1和Eureka2。以同样的方式搭建Zuul网关双节点Zuul1和Zuul2，提供8080访问端口提供服务。基础服务搭建完成后，部署数据清洗服务Data ETL双节点，其余业务系统采用同样方式部署双节点^[10]。

打包部署完成后，在任意两台2核8 GB服务器上部署Nginx节点，对外提供80端口，通过反向代理指向Zuul网关的8080端口，其余端口由防火墙控制拦截，防止集群被恶意攻击。

5 结论(Conclusion)

本方案采用物联网边缘计算与云原生的核心服务，构建

了一套冷链系统,通过Spring Cloud微服务架构和Docker容器化技术,使用Kubernetes、Jenkins等DevOps工具,实现系统的高效部署和快速增缩节点。该系统具有高性能、高稳定性和高扩容性,为物联网场景提供了一套安全、弹性、高效和稳定的解决方案。

参考文献(References)

- [1] 李中雷,陈杨.NB-IoT窄带物联网设备管理平台云原生多活研究与实践[J].广东通信技术,2021(11):26-31.
- [2] 孙其博,刘杰,黎彝,等.物联网:概念、架构与关键技术研究综述[J].北京邮电大学学报,2010,33(03):1-9.
- [3] International Telecommunication Union. Internet Reports 2005:The Internet of things[R]. Geneva:ITU, 2005.
- [4] 翟永超.Spring Cloud微服务实战[M].北京:电子工业出版社,2017:61-84.
- [5] 游灿虹.服务器集群负载均衡算法分析[J].现代计算机(专业版),2020(11):8-12.

(上接第44页)

- [2] 陈金广,马玲叶,马丽丽.求解作业车间调度问题的改进遗传算法[J].计算机系统应用, 2021,30(5):190-195.
- [3] 孙超平,杨平,李凯.考虑外包的平行机调度问题的多目标遗传算法[J].中国机械工程, 2014,25(23):3174.
- [4] HEBRARD E, HUGUET M J, JOZEFOWIEZ N, et al. Approximation of the parallel machine scheduling problem with additional unit resources[J]. Discrete Applied Mathematics, 2016, 215:126-135.
- [5] GAREY M R, JOHNSON D S. Complexity results for multiprocessor scheduling under resource constraints[J]. SIAM Journal on Computing, 1975, 4(4):397-411.

(上接第53页)

参考文献(References)

- [1] 董昌坤.基于HDFS的分布式云存储系统的设计与实现[D].北京:北京邮电大学,2013.
- [2] AISLING O A, JURATE D B, ROY D S B. 'Big data', Hadoop and cloud computing in genomics[J]. Journal of Biomedical Informatics, 2013, 46(5):774-781.
- [3] 袁志伟.在线学习行为日志数据挖掘算法优化[J].自动化与仪器仪表,2021(08):65-68.
- [4] 杨河山,张世明,曹小鹏,等.基于Hadoop分布式文件系统的地震勘探大数据样本采集及存储优化[J].油气地质与采收率,2022,29(1):121-127.
- [5] 袁斯琅.基于Hadoop与SSM的大数据云存储平台设计与实现[D].广州:广东工业大学,2018.
- [6] 薛荷.大数据存储优化及快速检索技术研究[D].成都:电子科技大学,

- [6] 卢莎莎,肖海力,王小宁.容器技术在高性能计算环境中的应用[J].数据与计算发展前沿,2021,3(6):118-126.
- [7] 浙江大学SEL实验室.Docker容器与容器云[M].2版.北京:人民邮电出版社,2016:261-368.
- [8] 刘殊旸,张曼怡,曹强.Docker动态调度算法的设计与实现[J].计算机工程与科学,2018,40(12):2112-2118.
- [9] 程冠杰,黄铮杰,邓水光.基于区块链与边缘计算的物联网数据管理[J].物联网学报,2020,4(02):1-9.
- [10] 戴启铭,毛润丰,黄璜,等.DevSecOps:DevOps下实现持续安全的实践探索[J].软件学报,2021,32(10):3014-3035.

作者简介:

郑响萍(1982-),女,硕士,实验师.研究领域:教育信息化管理,软件开发和信息服务.

蔡海军(1982-),男,硕士,实验师.研究领域:教育管理,大数据.

- [6] SEVASTIANOV S V, WOEGINGER G J. MAKESPAN minimization in open shops: A polynomial time approximation scheme[J]. Mathematical Programming, 1998, 82(1):191-198.
- [7] EDIS E B, OGUZ C, OZKARAHAN I. Parallel machine scheduling with additional resources: Notation,classification, models and solution methods[J]. European Journal of Operational Research, 2013, 230(3):449-463.

作者简介:

李志林(1995-),女,硕士生.研究领域:运筹学与应用,数学实验,智能算法.

技大学,2018.

- [7] 陈磊,吴晓晖.基于Hadoop的分布式集群大数据动态存储系统设计[J].中国电子科学研究院学报,2019,14(6):593-598.
- [8] 张姣娜.基于云计算的数据存储技术研究[D].西安:西安电子科技大学,2017.
- [9] 陈东辉,曾乐,梁中军,等.基于HBase的气象地面分钟数据分布式存储系统[J].计算机应用,2014,34(09):2617-2621.

作者简介:

蔡春花(1995-),女,硕士,助教.研究领域:大数据技术与应用.

黄思远(2000-),男,本科生.研究领域:深度学习.

高继梅(1984-),女,硕士,副教授.研究领域:大数据技术与应用.