



## 云原生边缘计算架构分析

张琦<sup>1</sup>, 贾玄<sup>2</sup>, 张森<sup>1</sup>, 殷薇<sup>1</sup>, 欧争光<sup>1</sup>, 王烽<sup>1</sup>

(1. 华为技术有限公司, 广东 深圳 518129; 2. 中国移动通信有限公司研究院, 北京 100055)

**摘要:** 为了解决日趋复杂的边缘应用管理带来的问题, 在分析业界与学术界多个边缘计算项目的架构后, 阐述了一种基于 Kubernetes 架构的云原生边缘计算系统 KubeEdge。KubeEdge 本质是将 Kubernetes 的云原生微服务管理能力延伸到边缘, 将云原生的生态和开发体验延伸到边缘, 针对上层开发者提供统一的开发、部署、管理视图, 并增强了离线运行能力、边云协同能力和边边协同能力。最后, 给出了在智慧园区、工业互联网、互动直播、自动驾驶等场景下 KubeEdge 的实际应用案例以及未来在边缘应用安全、数据隐私和可信以及边缘 AI 方向的研究工作。

**关键词:** 边缘计算; 云原生; 边云协同; 边边协同

**中图分类号:**

**文献标识码:** A

**doi:** 10.11959/j.issn.1000-0801.2019255

## Analysis of cloud native edge computing architecture

ZHANG Qi<sup>1</sup>, JIA Xuan<sup>2</sup>, ZHANG Sen<sup>1</sup>, YIN Wei<sup>1</sup>, OU Zhengguang<sup>1</sup>, WANG Feng<sup>1</sup>

1. Huawei Technologies Co., Ltd., Shenzhen 518129, China

2. China Mobile Research Institute, Beijing 100055, China

**Abstract:** In order to solve the problems brought by the increasingly complex edge application management, the architecture of multiple edge computing projects in the industry was analyzed, and KubeEdge, as a cloud-based edge computing system based on Kubnetes architecture was expounded. KubeEdge essentially extended Kubernetes' cloud-native micro-service management capabilities to the edge, extended the cloud's native ecosystem and development experience to the edge, provided unified development, deployment, management views for upper-level developers, and enhanced off-line capabilities, collaboration capabilities crossing edge to cloud as well as edge to edge. In the end, practical applications of KubeEdge in smart campus, industrial internet, interactive live broadcast, autopilot, etc. were given, as well as future research work on edge application security, data privacy and credibility, and edge AI direction.

**Key words:** edge computing, cloud native, edge cloud collaboration, edge collaboration

### 1 引言

随着移动终端及物联网技术的发展和普及应

用, 当前万物互联的世界中产生的数据量将以爆炸式的增长达到 ZB 级别, 以云计算模型为核心的集中式大数据处理模型已经不能满足大带宽、

收稿日期: 2019-08-20; 修回日期: 2019-10-19

低时延、数据隐私安全要求高的业务需要,为此,以边缘计算模型为核心的边缘式大数据处理应运而生<sup>[1]</sup>。

当前边缘计算技术在智能工业、智慧医疗、智慧城市、智能交通与车联网、智慧物流及智能家居等领域已经有了初步的尝试和应用,Gartner预测在2022年50%的大型企业将应用边缘计算技术。随着边缘计算应用场景的日益增加和边缘设备产生的数据量的日益增多,边缘业务的复杂性日趋增高,为边缘应用的管理带来一系列挑战。

(1) 边缘应用开发、部署的便捷性要求增高,以满足边缘业务的快速发展和变化。

(2) 边缘应用需要有完善的微服务治理能力,以满足日趋复杂的边缘业务模型。

(3) 边云、边边的协同成为边缘应用的基本要求,以满足海量边缘数据的处理。

为了解决上述问题,本文将阐述一种边云、边边协同的CloudNative模式的边缘计算系统架构。

## 2 背景

### 2.1 发展现状

从边缘计算的发展历程来看,2015年前其处于原始技术积累阶段,Cloudlet、雾计算、海云计算被相继提出;2015–2017年期间,边缘计算快速发展,工业界发表了一系列的白皮书并成立了相关工作组及联盟,如欧洲电信标准化协会(ETSI)定义的多接入边缘计算(multi-access edge computing, MEC),思科、ARM、戴尔、英特尔、微软和普林斯顿大学成立的OpenFog联盟,华为技术有限公司、中国科学院沈阳自动化研究所、中国信息通信研究院、英特尔、ARM等成立的边缘计算产业联盟(Edge Computing Consortium)等。2018年是边缘计算发展过程中的重要节点,在这一年边缘计算被推向前台。

### 2.2 学术界边缘计算平台研究

学术界关于边缘计算平台的研究主要有如下两个项目:Picasso和Discovery。

#### (1) Guifi Picasso

Guifi项目由英国剑桥大学发起,目标是为欧洲边远山区的民众搭建社区网络(community network),Picasso是Guifi项目的轻量级边缘计算平台部分。由市民、商户、学校、组织利用一些商用路由器以及Wi-Fi、光纤搭建了一个开放、互联的网络,并利用计算、存储资源构建了一个完整的去中心化云服务。

Picasso主要关注基于服务质量(quality of service, QoS)的服务调度、扩展和服务发现。使用了服务发现工具avahi(mdns/dns-sd)以及基于gossip协议最终一致性的消息组件serf(集群管理、消息传递),架构如图1所示。

#### (2) Discovery Initiative

Discovery项目希望构建一个去中心化的基础设施即服务(infrastructure as a service, IaaS)管理平台,以应对管理地理离散的边缘计算节点的挑战。该项目基于OpenStack改造,通过替换集中式的数据库与消息组件为类P2P的redis cluster和zero MQ来支持当地理离散的节点增加时,资源管理性能的水平扩展。Initiative架构如图2所示。

### 2.3 边缘计算开源项目

边缘计算得到了技术社区的大力支持,其中具有代表性的是以下4个开源组织及项目。

#### (1) LF Edge

LF Edge是Linux基金会的一个伞形组织,旨在建立一个开放、可互操作的边缘计算框架,它最初由5个项目组成,包括Akraino Edge Stack、EdgeX Foundry、Home Edge、Open Glossary of Edge Computing和Project EVE。LF Edge的边缘侧项EdgeX Foundry具备边缘数据收集云端汇聚功能,但当前并未关注边缘与云在应用开发、部

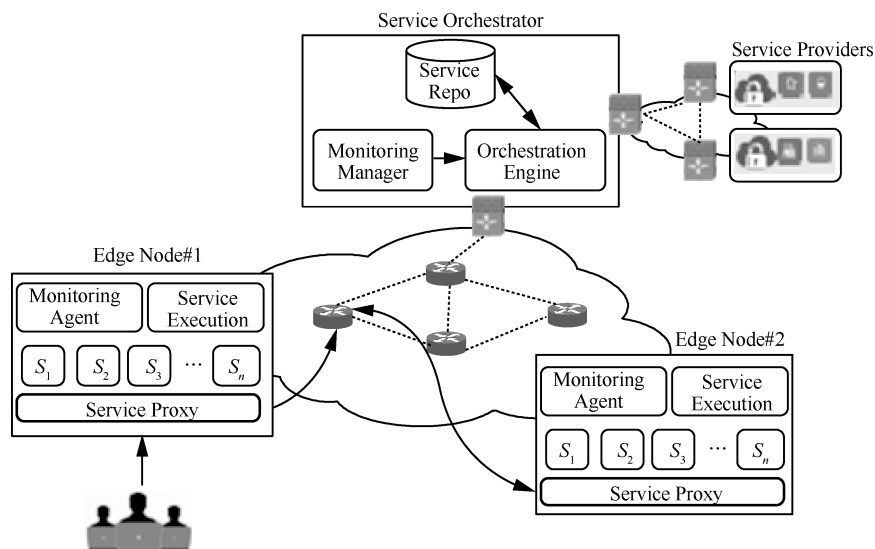


图 1 Picasso 架构

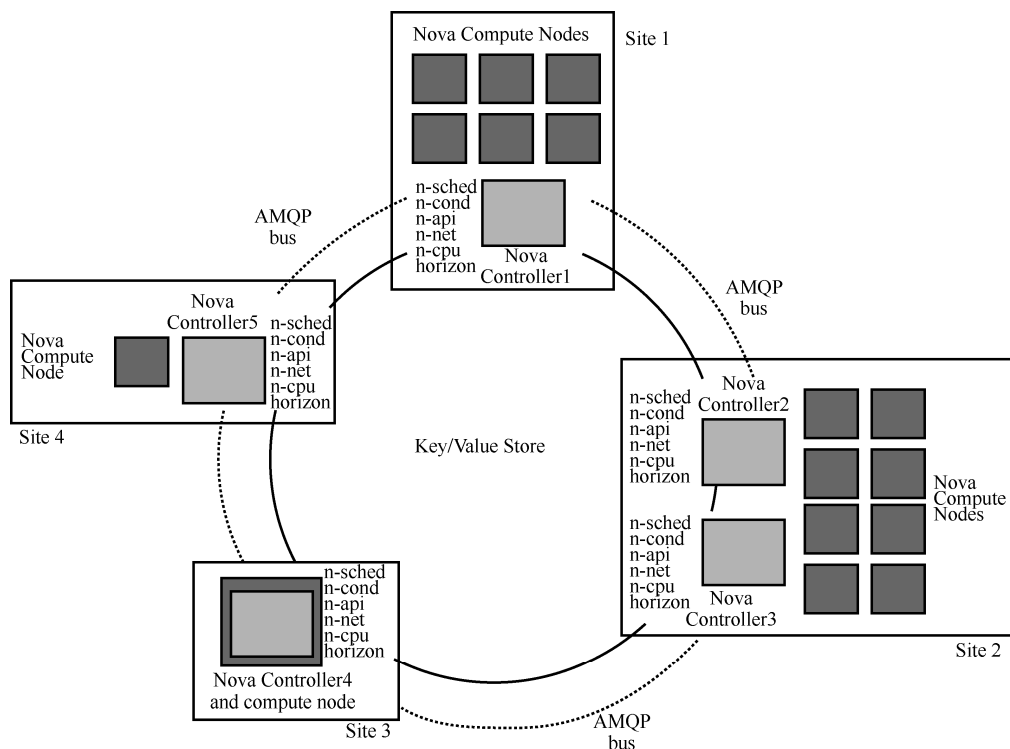


图 2 Initiative 架构

署、管理、调度上的协同，不能很好地解决云与端的生态割裂。EdgeX Foundry 架构如图 3 所示。

## (2) Eclipse IoT

2016 年，Eclipse 基金会开源多个物联网项目，如图 4 所示，包括 Eclipse Kura、Eclipse Paho、Eclipse OM2M 以及 Eclipse SmartHome。Eclipse

将物联网视为包括三层互联的软件栈。

- 设备应用程序的堆栈；
- 采集数据信息聚合的网关；
- 物联网平台后端的软件栈，通常包括云存储以及数据分析服务。

Eclipse IoT 当前项目提供了大量的端侧与边

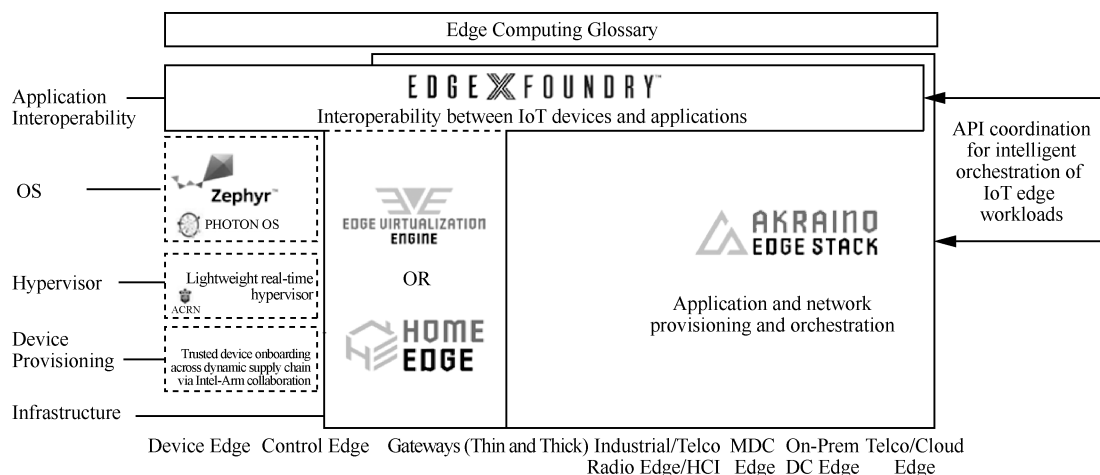


图3 EdgeX Foundry 架构

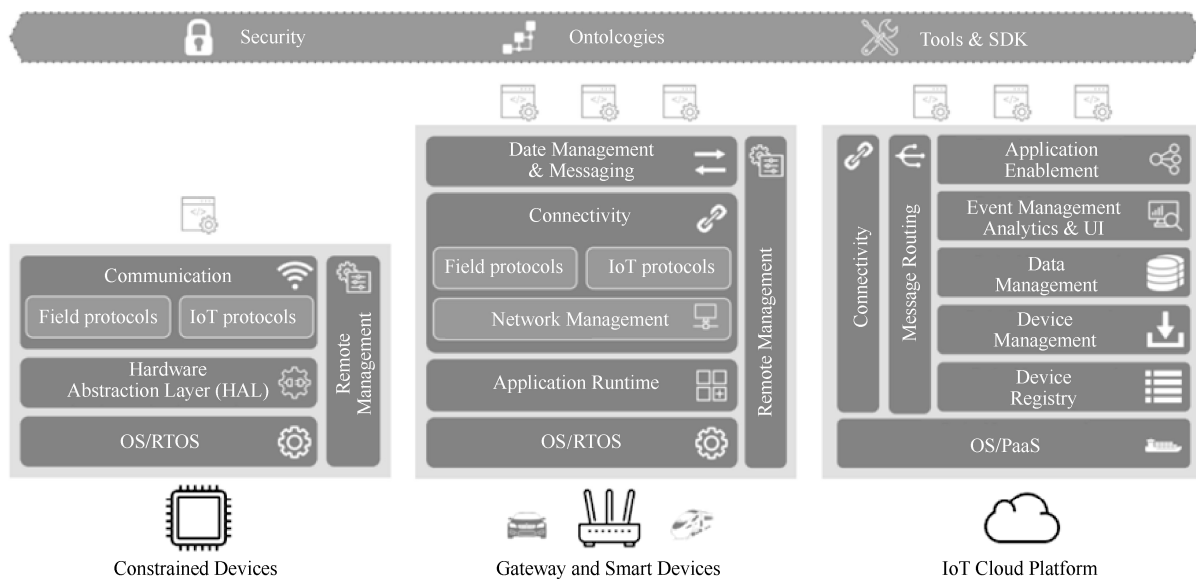


图4 Eclipse IoT 架构

缘设备的开发框架与中间件，边缘侧的项目 *iofog* 更偏重嵌入式基础设施平台及开发套件，而对端云一体的生态以及边云应用协同的开发、部署、管理、调度涉及不多，难以利用云端的强大计算能力和丰富服务。

### (3) ETSI

欧洲电信标准化协会 ETSI (European Telecommunications Standards Institute) 定义 MEC 为 *multi-access edge computing*, 打造一个融合电信移动网络和云技术的边缘生态环境。ETSI MEC 提出基于网络功能虚拟化 (*network function virtu-*

*alization*, *NFV*) 的边缘计算架构如图 5 所示。

未来 5G 核心网都会运行在 *NFV* 中, 并且 5G *UPF* 提供相应的分流功能。基于这样的思路, ETSI 提出在 *NFV* 上构造 MEC 的做法, 对原有 *NFV* 体系架构进行补充。其优势是构造边缘计算的运营商和供应商能够很好地达成语言上的共识, 缺点是未充分考虑最终用户的需求和使用习惯。

### (4) CNCF

2018 年 11 月, *KubeCon* 上正式公布了新的基于 *Kubernetes* 的生态系统 *KubeEdge*, 将 *Kubernetes* 生态系统从云端扩展到边缘, 将在本

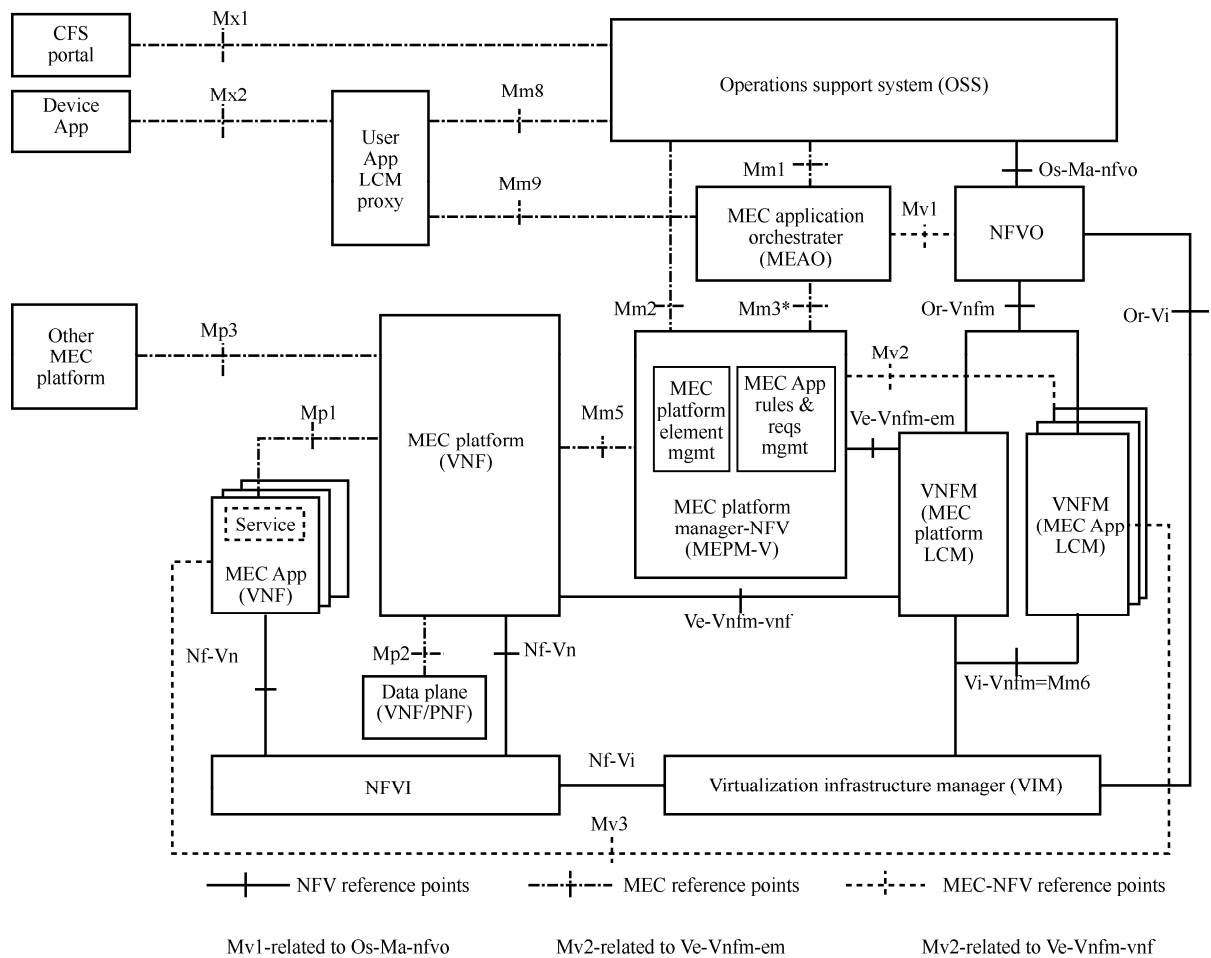


图5 ETSI MEC 架构

文第3节详细阐述。

### 3 云原生的边缘计算架构

#### 3.1 整体架构和生态支持

Kubernetes是Google在2014年开源的容器编排调度系统，致力于支撑云原生应用的部署编排和为微服务运行提供所必需的基础设施。Kubernet的Edge Computing Working Group聚焦云原生技术和Kubernetes在边缘计算场景下的技术方案、参考架构已经构建端到端的验证设计并提供系统集成样例。本文所描述的云原生边缘计算架构便是该组织中的边缘计算参考架构，该架构由华为公司提出，于2018年11月宣布以KubeEdge为名开源，现已是CNCF官方项目之一。

KubeEdge基于Kubernetes的架构体系并针对边缘场景提供了诸如离线运行能力、边云协同能力等多种特殊能力的支持，将云原生的生态和开发体验延伸到边缘，面向开发者提供统一的开发、部署、管理视图，屏蔽边缘和云端的差异。其整体架构如图6所示。

##### (1) 云端组件

为了兼容Kubernetes生态，云端组件采用标准的Kubernetes架构模式进行构建，架构核心基于Kubernetes的“预期状态模式”，处理步骤如下。

- 用户通过kubectll命令行发出对目标对象预期状态的指令；
- Kubernetes的API Server接收到该指令，调度器会对其进行调度并将用户的指令分

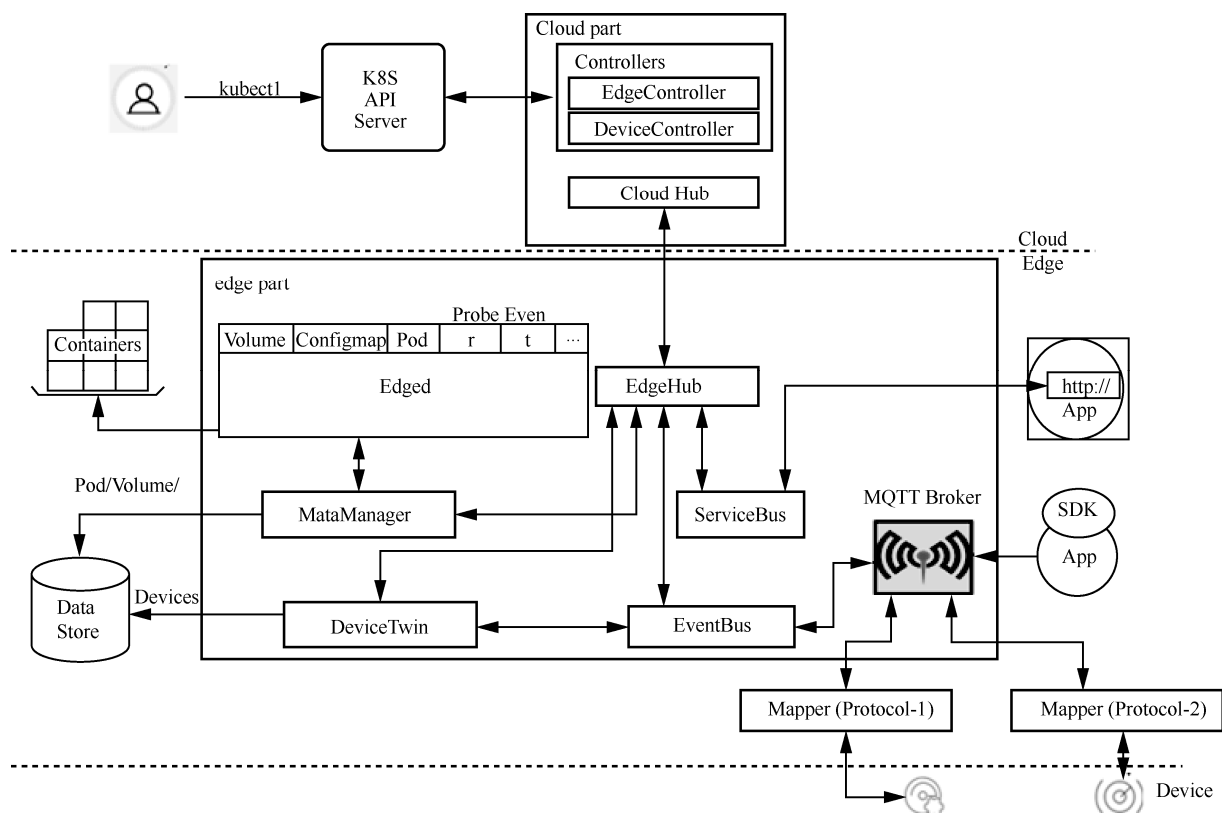


图6 KubeEdge 架构

解为一系列新的带有“预期状态”的子对象；

- 新增加 EdgeController 和 DeviceController 通过 API Server 的“watch”接口订阅其所关心的对象，发现有新增对象或对象状态有更新则继续后续处理；
- Controller 将监听到的对象变动送给 CloudHub 组件，该组件维护一个边缘侧通过 WebSocket 协议连接的列表，结合对象中的目的节点选择性地将该对象的元数据送至相应的目的边缘节点。

## (2) 边缘侧组件

为了进一步降低边缘侧的资源占用、出问题概率和维护难度，边缘侧组件本着“简单至上”的原则进行设计。所有的核心组件都运行于相同的进程内，将不同的功能组件定义为不同的模块 (module)，通过名为 Beehive 的框架统一管理，使用轻量级的 go 语言协程运行各个 module。

Beehive 框架提供了模块的注册、发现、通信能力，在通信能力中又提供了同步、异步、单播、组播的能力以及通过 go channel 提供了进程内跨协程的通信能力和通过 Unix Domain Socket 提供了模块间跨进程的通信能力。

整体的边缘侧处理流程如下所示。

- EdgeHub 组件启动后主动发起与云上服务的连接，通过 WebSocket 协议构建一条长连接，接收云上下发消息并上报边缘的数据和状态。
- 接收到云上下发的某资源对象的元数据后，EdgeHub 将该数据送给 MetaManager 模块；MetaManager 模块统一管理云上下发的管理对象的元数据，使用 SQLite 数据库将数据进行持久化，并向其他动作执行模块提供查询接口，当对象元数据、状态等发生变化后，更新数据库中的相关数据。



- Edged 模块负责边缘端的应用管理, 根据收到的应用对象元数据, 通过 Docker engine 的接口启动相应的 Docker 容器, 并管理容器的网络、存储、配置信息等。
- DeviceTwin 维护边缘侧的 DeviceTwin 数据, 端侧设备的数据通过 Mapper 收集至该模块, DeviceTwin 负责将数据进行持久化并对云上和边缘侧的其他应用提供标准化的数据访问接口。
- EventBus 负责维护一个内置 Mqtt Broker 的数据转发, 并且可以使用 Bridge 模式对接一个外置的 Mqtt Broker。边缘侧其他应用产生的数据或收集到的端侧设备的数据经 EventBus 转发至云上或其他边缘侧模块。同时, Edge-core 中的其他模块需要与边缘侧其他应用以 Mqtt 消息的方式(推荐的边缘侧消息交互模式)进行交互的时候都需通过该模块进行。
- ServiceBus 的定位同 EventBus, 负责通过 REST 服务的形式联通云端和边缘侧的应用。当边缘侧的 REST 服务没有公网 IP 地址时, 云上的 REST 请求可以通过 KubeEdge 提供的 WebSocket 连接传送至边

缘侧, ServiceBus 负责将该请求转发至正确的边缘应用, 而后将 Response 返回至云上相应的请求发送方处。

- Mapper 和 DeviceTwin 共同构成了物理设备的世界到数字世界的映射。Mapper 通过不同设备协议与其相连并将数据按照 DeviceTwin 的格式转换为 DeviceTwin 模型数据, 通过 Mqtt Broker 将数据送入 DeviceTwin 模块供其他应用访问。

### 3.2 支持复杂的部署模型

Kubernetes 在多年的发展中已经积累和沉淀了一些对云原生应用的编排部署模型, KubeEdge 将这些编排部署能力延伸到边缘侧, 支持边缘侧日益复杂的业务和高可用性的要求。

Kubernetes 的整体模型如图 7 所示, 在图 7 中可以表现大多数的云原生编排部署能力。

#### (1) 双机热备

在一些存储持久化的场景中, 例如 MySQL 双机热备的场景, 利用 statefulset 的机制, 在应用重新调度后 MySQL 应用还能被调度到原有的节点之上, 保证访问同样的存储。

#### (2) 多机多活互备

可以利用 replicaset 创建多个副本, replicaset

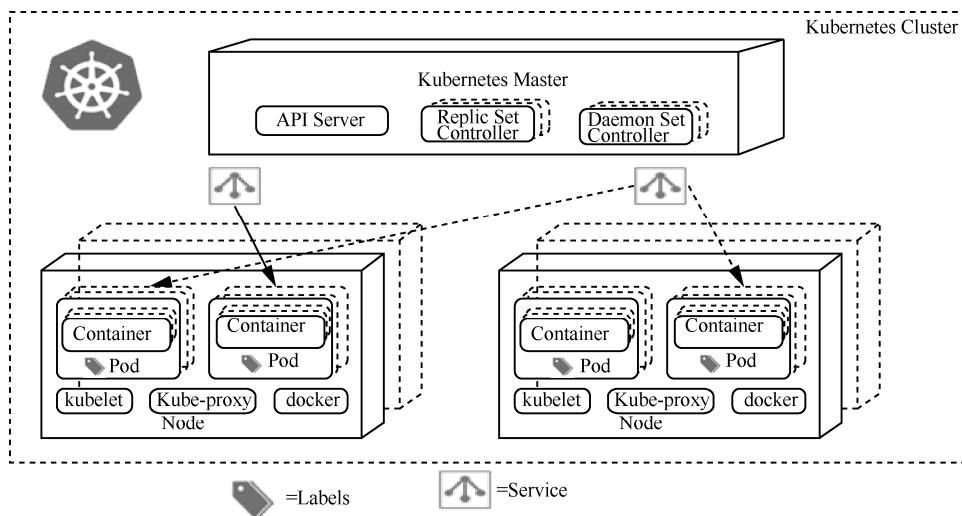


图 7 Kubernetes 对象模型

可以维持设定副本数的应用同时运行。

(3) 有关联的应用同节点部署以提升应用间交互效率

使用 POD 亲和性调度规则, 可以把相关的应用部署在相同的主机上以提升通信性能。

(4) 同一应用的不同实例跨节点部署以提升可用性

利用 POD 反亲和性特性, 可以选择同一应用部署的不同实例分散在不同节点上进行部署。

(5) 依据边缘节点的不同属性将应用部署于不同分组中

在很多场景中依据边缘节点的地理位置、设备类型、功能属性、性能等分属于不同的分组, 打上不同的标签, 在应用部署时利用 node selector 的能力将应用部署在对应 label 上的边缘节点之上。

(6) 定义独立于节点的应用部署以实现满足条件的新边缘节点上线后自动安装应用

在应用部署时, 应用根据节点的 label 去将应用部署在对应的节点之上, 当新的节点上线后, 可以将其打上同样的 label, 节点上线之后 k8s 系统可以立刻将应用自动部署在新上线的边缘节点之上。

### 3.3 使用 edgemesh 支持跨越边界的微服务访问

随着微服务框架的流行和边缘节点计算和网络能力的增强, 把部分服务部署到边缘处理设备

产生的数据可以解决时延的问题, 但由于安全等原因, 大多数服务仍然运行在云上, 云跟边、边跟边之间的协同是必须要解决的问题。edgemesh 微服务框架如图 8 所示, edgemesh 主要解决了两个问题: 由跨越了边、云而组成的大型分布式系统中的微服务之间互相发现和访问; 在微服务互相访问的过程中对访问过程进行治理、控制从而提升大型分布式系统整体的可用性。

#### (1) 边边协同

边边协同原理如图 9 所示。

- 在 k8s 中创建 Service, 设置 label;
- edgecontroller 观测到 Service 的事件, 同步到 edgehub, 经 metamanager 保存到 SQLite;
- 用户应用 A 跨接点请求用户应用 B 时, 流量通过 iptables 规则全部劫持到 edgemesh;
- edgemesh 解析应用层协议, 调用 metamanager 查询 Service/Endpoint/Pod 等相关信息, 根据负载均衡等策略选择目的地址, 封装请求为 “beehive/pkg/core/model”。Message 并转发请求到目的节点的 edgemesh;
- 目的节点 edgemesh 解析 “beehive/pkg/core/model”。Message 并根据配置的 QPS 策略处理请求, 转发请求到用户边缘应用 B。

#### (2) 边云协同

边云协同原理如图 10 所示。

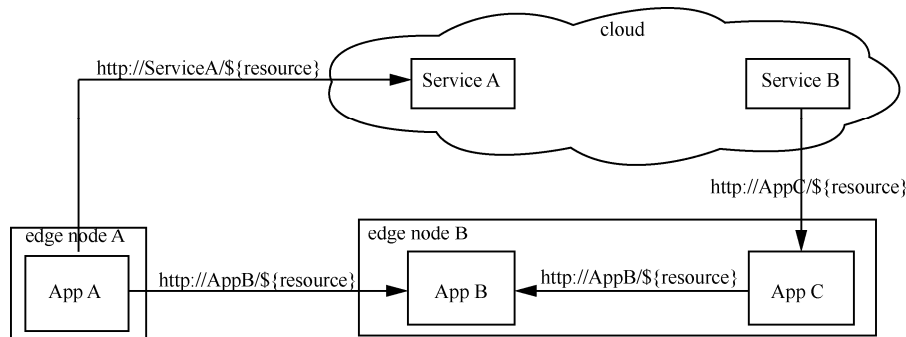


图 8 edgemesh 微服务框架



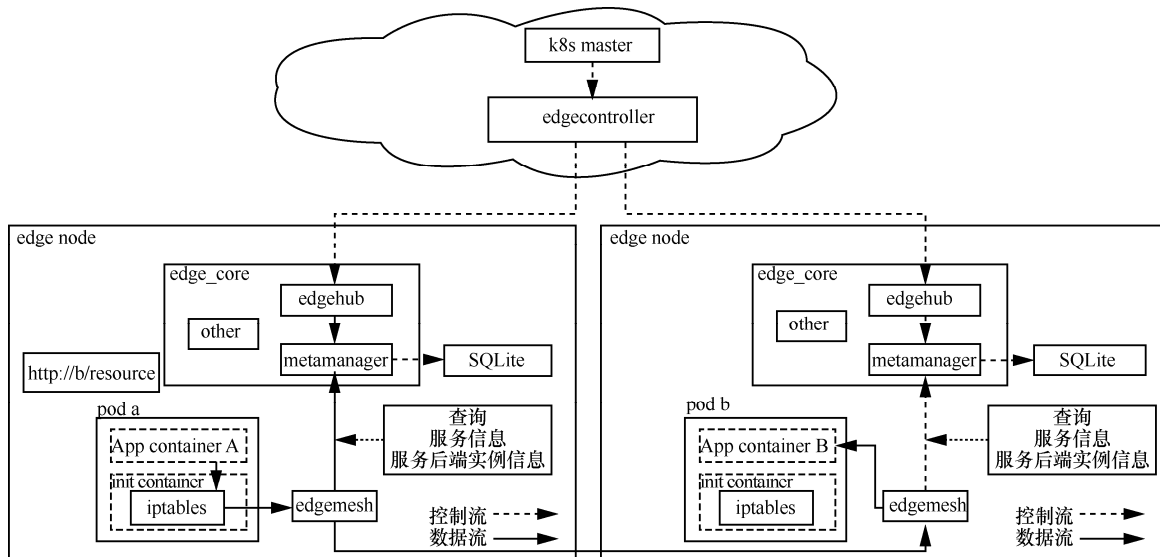


图9 边边协同

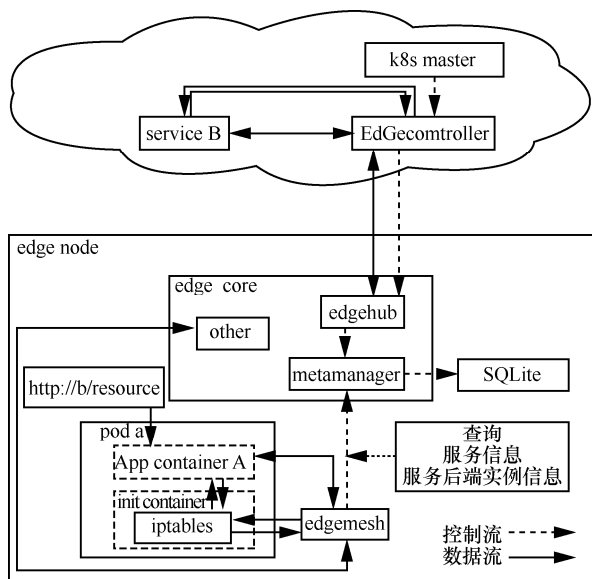


图10 边云协同

- 在 k8s 中创建 Service，设置 label；
- edgecontroller 观测到 Service 的事件，同步到 edgehub，经 metamanager 保存到 SQLite；
- 用户应用 A 跨接点请求用户云上服务 B 时，流量通过 iptables 规则全部劫持到 edgemesher；
- edgemesher 解析应用层协议，调用 metamanager 查询 Service/Endpoint/Pod 等

相关信息，根据负载均衡等策略选择目的地址，封装请求为 “beehive/pkg/core/model”。Message 并转发到 edgecontroller；

- edgecontroller 解析 “beehive/pkg/core/model”，Message 根据配置的 QPS 策略处理请求，转发请求到用户云上服务 B。

## 4 应用情况介绍

### 4.1 智慧园区

智慧园区项目基于华为云智能边缘平台 IEF (intelligent EdgeFabric, KubeEdge 的商业版) 服务在边缘侧部署人脸识别、人员轨迹分析、事件报警管理等边缘应用，大幅度降低了视频上云对网络带宽的诉求。

智慧园区如图 11 所示，视频分析算法以容器的方式快速部署到边缘节点，并同云上的人脸检索等服务对接形成边云协同的智能视频分析服务。基于 IEF 可以实现边缘算法的动态更新及灰度升级。

### 4.2 工业互联网

某合成纤维领域的高新技术企业使用 IEF 实现生产状态的实时监控和产品质量的实时预测，提升产品质检的检测效率，提高产品质量，减少

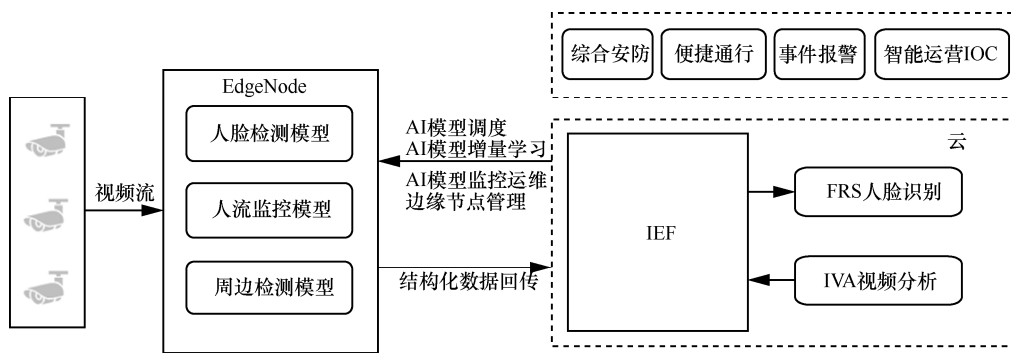


图 11 智慧园区

生产线故障率。

工业互联网如图 12 所示，将工业视觉质检应用部署在边缘节点以满足工业级实时性，同时部署数据预处理应用将数据脱敏后上传到云上训练，并实时更新边缘模型，提高推理精度。

#### 4.3 互动直播

互动直播如图 13 所示，某视频直播企业在 CDN 侧通过华为智能边缘平台 IEF 部署视频转码、渲染、切片等多个边缘微服务，对数据进行预处理，减少向中心服务器传输的数据量，降低网络传输成本；为了提升边缘微服务的高可用性，通过 IEF 的边边协同能力实现边缘微服务跨边缘

站点的负载均衡和自动伸缩。

#### 4.4 自动驾驶

自动驾驶如图 14 所示，自动驾驶是典型的云边端协同架构。汽车车体上拥有各种传感器（如图像传感器等），通过 5G 信号传输到就近的边缘服务器上，对数据转换和初步分析、推理，处理后的关键数据上传到云端，进行进一步推理预测和训练。

依赖 5G 和边缘计算，实现了车辆的自动驾驶和车联网。云边端的协同，是时延、数据量、计算量的三重均衡结果。采用云原生架构，可更加灵活地分配计算量和数据量，并满足时延的需求。

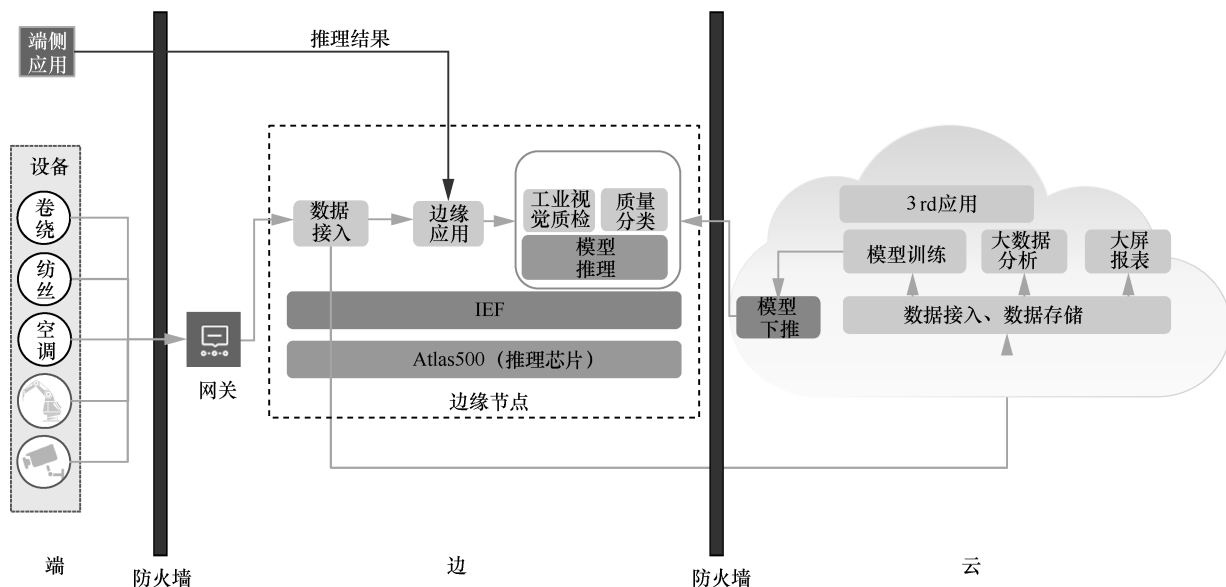


图 12 工业互联网

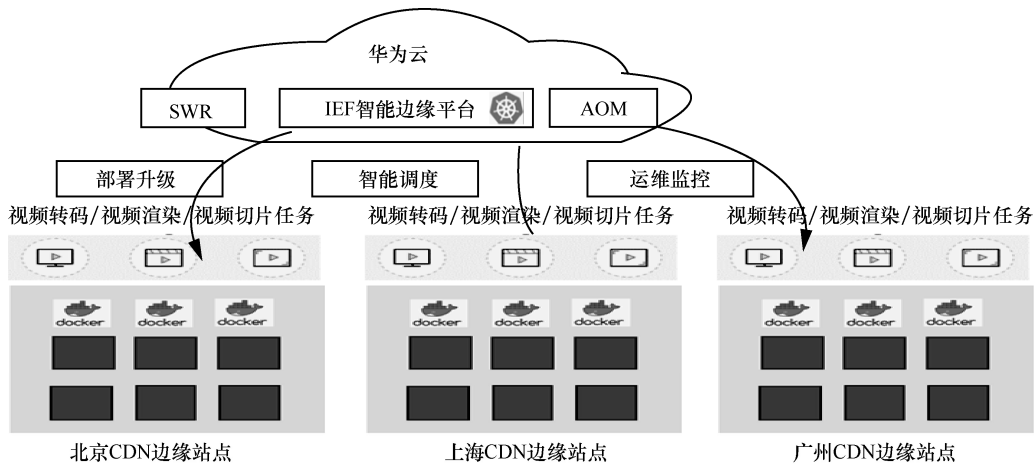


图 13 互动直播

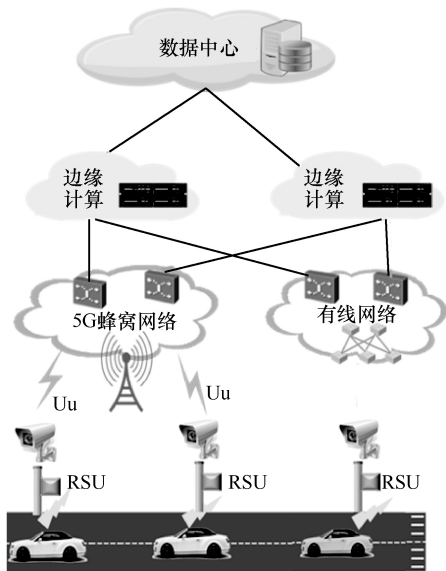


图 14 自动驾驶

保护,近年兴起 Zero Trust 网络和 SDP (software defined perimeter, 软件定义边界),或许是适于边缘节点的一个方向<sup>[2-3]</sup>。

#### (2) 数据隐私和可信问题<sup>[6]</sup>

一种尝试是可信计算,在 TEE (如 ARM Trustzone) 中对数据进行加密再传输,应用运行在 TEE 中进行解密、处理;另一种可行的尝试是多方计算 MPC 和同态加密。

#### (3) 边缘 AI

当前流行云上训练、边缘推理模式,实际上由于数据隐私和网络带宽问题,大量数据分布在边缘,分布式机器学习或联邦学习<sup>[4-6]</sup>在边缘对数据进行训练、中心进行协调,可能成为一个重要方向。

## 5 未来的工作

边缘计算方兴未艾,未来研究方向广阔,具体如下。

### (1) 安全

传统应用运行在数据中心,通过一个统一的入口对外提供服务,而边缘应用部署在一个不可信的环境中,这种方式给应用和数据安全带来了极大的挑战。边缘节点与中心云通过公有网络或专线联通,采用防火墙、安全组等传统方式进行

## 6 结束语

本文阐述的云原生边缘计算架构将云原生应用无缝延伸到靠近数据产生位置的边缘侧,实现了边云、边边之间的应用协同和统一微服务治理,从而应对日趋增高的边缘应用复杂性对应用开发、部署、管理、协作维度带来的挑战。本文结合实际应用案例完整地描述了云原生边缘计算的架构,同时给出了面向边缘应用安全、数据隐私和可信以及边缘 AI 等下一步的研究方向。

## 参考文献:

- [1] 施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(5):907.  
WEI W S, SUN H, CAO J, et al. Edge computing— an emerging computing model for the internet of everything era[J]. Journal of Computer Research and Development, 2017, 54(5): 907.
- [2] PUTHAL D, NEPAL S, RANJAN R, et al. Threats to networking cloud and edge datacenters in the internet of things[J]. IEEE Cloud Computing, 2016, 3(3): 64–71.
- [3] PUTHAL D, MOHANTY S P, NANDAP, et al. Building security perimeters to protect network systems against cyber threats [J]. IEEE Consumer Electronics Magazine, 2017, 6(4): 24–27.
- [4] MCMAHAN B, RAMAGE D. Federated learning□: collaborative machine learning without centralized training data[J]. Post, 2017: 1.
- [5] LI P, LI T, YE H, et al. Privacy-preserving machine learning with multiple data providers[J]. Future Generation Computer Systems, 2018, 87: 341-350.
- [6] SMITH V, CHIANG C K, SANJABI M, et al. Federated multi-task learning[Z]. 2017: 4424-4434.

## [作者简介]

张琦(1982- ), 男, 华为云智能边缘云服务首席架构师, 主要研究方向为云原生技术、边缘计算等。

贾玄(1987- ), 男, 中国移动研究院边缘计算架构师, 主要研究方向为边缘计算平台、边缘计算与 5G 网络协同等。

张森(1982- ), 男, 华为云智能边缘云服务技术专家, 主要研究方向为边缘智能边缘云平台、端边云协同创新等。

殷薇(1982- ), 女, 华为云智能边缘云服务架构师, 主要研究方向为边缘计算平台及相关应用。

欧争光(1987- ), 男, 华为云智能边缘云服务高级工程师, 主要研究方向为边缘计算平台、端边云协同等。

王烽(1982- ), 男, 博士, 华为云智能边缘云服务架构师, 主要研究方向为边缘计算架构、边缘 AI 等。