

基于边缘计算的云原生机器人系统

王大伟¹, 王卓¹, 王鹏¹, 王志刚¹, 王海涛²

(1. 英特尔(中国)研究中心有限公司, 北京 100490; 2. 英特尔亚太研发有限公司, 上海 200241)

摘要: 随着人工智能的飞速发展, 全球机器人市场在持续扩大, 机器人的能力从执行固定的操作发展到自主感知、理解和决策。但要实现机器人的大规模应用, 需要机器人在有限功耗的限制下, 具备强大的计算能力和低廉的部署成本。利用边缘计算提供性价比更高的服务, 增强机器人本体的计算能力, 实现规模化部署是实现这一目标的关键。分析了实现边缘增强的机器人系统面临的挑战, 提出了基于边缘计算的云原生机器人系统, 并探讨了实现该系统的4种可行的技术方案。提出的系统是由基于机器人本体的智能系统向云-边-端融合的多机器人协同的智能系统发展的趋势, 也是实现机器人大规模应用的关键技术。

关键词: 机器人; 机器人4.0; 云原生; 边缘计算; 机器人操作系统

中图分类号: TP242.6

文献标识码: A

doi: 10.11959/j.issn.2096-6652.202030

Cloud native robot system based on edge computing

WANG Dawei¹, WANG Zhuo¹, WANG Peng¹, WANG Zhigang¹, WANG Haitao²

1. Intel China Research Center Co., Ltd., Beijing 100190, China

2. Intel Asia-Pacific Research & Development Co., Ltd., Shanghai 200241, China

Abstract: With the rapid development of artificial intelligence, the global robot market continues to grow quickly, and the capabilities of robots have evolved from performing fixed operations to the ability to autonomously sense, understand and make decisions. However, to achieve large-scale application of robots, robots need to have powerful computing capabilities and low deployment costs under the constraints of limited power consumption. Using edge computing to provide more cost-effective services, enhance the computing power of the robot body, and achieve large-scale deployment is the key to achieving this goal. The challenges faced by robot systems with the edge enhancement were analyzed, the concept of cloud-native robot systems based on edge computing was proposed, and four feasible technical solutions for implementing the system were discussed. The cloud-native robot system is the inevitable direction for the development of robot systems from intelligent systems based on robot ontology to cloud-edge-end fusion multi-robot collaborative intelligent systems and the key technology for promoting the large-scale application of robots.

Key words: robot, robot 4.0, cloud native, edge computing, robot operating system

1 引言

当前, 全球机器人市场规模持续扩大, 工业机器人市场增速稳定, 服务机器人市场增速明显。2019年, 全球机器人市场规模达到294.1亿美元, 其中, 工业机器人159.2亿美元, 服务机器人94.6亿美元, 特种机器人40.3亿美元。2014年至2019年

的年平均增长率约为12.3%^[1]。在工业制造领域, 机械臂凭借其强大的负重能力和精准的抓取操作代替了工人的双手; 在物流领域, 智能仓储机器人和无人搬运车不断提高着运输效率; 在生活服务领域, 家用清洁机器人和服务机器人正成为许多家庭的私人保姆和小秘书。虽然服务机器人的整体销售额低于工业机器人, 但近几年一直维持着较高的年

收稿日期: 2020-07-30; 修回日期: 2020-08-31

通信作者: 王大伟, dawei.wang@intel.com

增长率。商用服务机器人在商场、银行、酒店、机场等应用场景有了更多的落地部署，主要提供导览、问询、送物等基础服务。同时，家用服务机器人也逐步进入千家万户，扫地机器人销量在家用服务机器人销量中占主要份额，成为目前家用服务机器人中的主导品类。

从技术发展和市场环境来看，机器人产业拥有很多发展推动力，如成熟的生态系统、人口老龄化趋势、更多的智能产品互联和智能家庭建设以及人工智能、自然语言理解能力的增强等。在以上几点的助推下，机器人产业将继续快速发展，但要达到大规模商用，还有很多难点需要解决，其中机器人对计算的需求尤为突出。推动机器人产业发展的 AI 技术，需要很强的计算能力，而机器人有限的计算能力限制了机器人的智能程度。同时，机器人市场没有形成规模，导致机器人价格依然很高，不能被广泛的市场用户所接受。

当前，机器人正处在机器人 3.0 时代^[2]，而参考文献[3-4]提出的机器人 4.0 时代则把机器人的大脑分布在从云到端的各个地方，并充分利用边缘计算提供性价比更高的服务，其给出了解决机器人计算需求的可行方案，为实现机器人的大规模部署提供了可能。边缘计算是近两年提出的计算模式概念，其直面云端数据中心之外的海量数据，同时，得益于 5G 的部署和 AI 研究的突飞猛进，其应用前景广阔，是多个产业升级的潜在发动机。本文将重点讨论机器人的计算系统，分析机器人未来发展对计算系统的挑战，以及为适应机器人应用以及大规模商用需求，机器人计算系统将向基于边缘计算的云原生机器人系统发展的趋势，进而提出可行的技术方案。

2 机器人计算平台面临的挑战

随着科技的飞速发展，人们对机器人的期望越来越高。机器人的终极目标是具备或者超越人类的各项能力。首先，这就要求机器人具备感知、推理和决策的能力，机器人必须具备对周围环境和环境中物体的感知能力，并且能够根据周边环境做出合适的响应。其次，机器人还需要具备交互的能力，与人类进行自然语言和行为的交互。此外，运动能力也是机器人所必备的，比如在地面移动以及操控物体。为了支撑运动能力，机器人需要具备对环境和自身位姿的感知能力。这些都是机器人需要具备的基本能力。与此同时，机器人还需要具有在不同

场景下以及同一场景随时间等变化的情况下，完成不同任务的能力，这些任务包括教育、娱乐、社交、工具服务、远程服务等。因此，机器人通常需要具备多种基本能力以及完成多种任务的能力，这对机器人的计算平台提出了非常大的挑战。

为此，笔者团队构建了开放的机器人参考平台，并在该机器人参考平台上以机器人自适应人机交互应用^[5-6]为样例，研究机器人不同的工作负载对计算平台的挑战。开放的机器人参考平台的软件协议栈如图 1 所示。在机器人硬件平台之上是基于 Linux 的操作系统，操作系统之上是开发工具层，包括英特尔公司推出的 OpenVINO SDK、OpenCL SDK、NCSDK 等。在此之上是机器人操作系统 (robot operating system, ROS)^[7]的运动功能模块层，包括同步定位与建图 (simultaneous localization and mapping, SLAM)、自动导航、运动规划、操纵等。感知层包括人脸识别、身份识别、人体识别、物体检测、表情识别和行为识别等。在感知层之上是自适应人机交互层，包括自适应身份识别、自适应物体识别和自适应情绪识别以及三维语义场景理解和个性化知识图谱。在应用层实现了智能提醒、寻找物体、日常行为检测和情绪控制等机器人应用。运动功能模块层、感知层、自适应人机交互层都是机器人平台的功能模块。

在机器人平台的功能模块中，既有传统的以计算机视觉为主的功能模块（如视觉 SLAM）、非 AI 计算的逻辑功能模块（如自动导航），也有各种基于神经网络的功能模块（如物体检测、人脸识别）等。机器人系统中各种功能模块对计算有不同的需求，如视觉 SLAM 需要对大量的特征点进行视觉处理，适合矢量并行计算，在 GPU 上具有更好的性能；各种基于深度神经网络的视觉感知模块以矩阵计算为主，在 AI 专用加速器 ASIC 上有更好的性能；运动导航等以标量计算为主的算法模块，通常在 CPU 上具有更好的性能；现场可编程门阵列 (field programmable gate array, FPGA) 在实时控制任务（比如传感器同步和驱动器控制等任务）中具有独特的优势，适用于运动控制任务，同时 FPGA 的空间计算能力也非常适用于对空间数据流的处理。因此，机器人需要一个异构的计算平台，以应对机器人多种任务的计算需求，达到最佳系统性能或者最佳能效比，例如 HERO 平台^[8]探讨了利用异构计算来加速机器人任务。

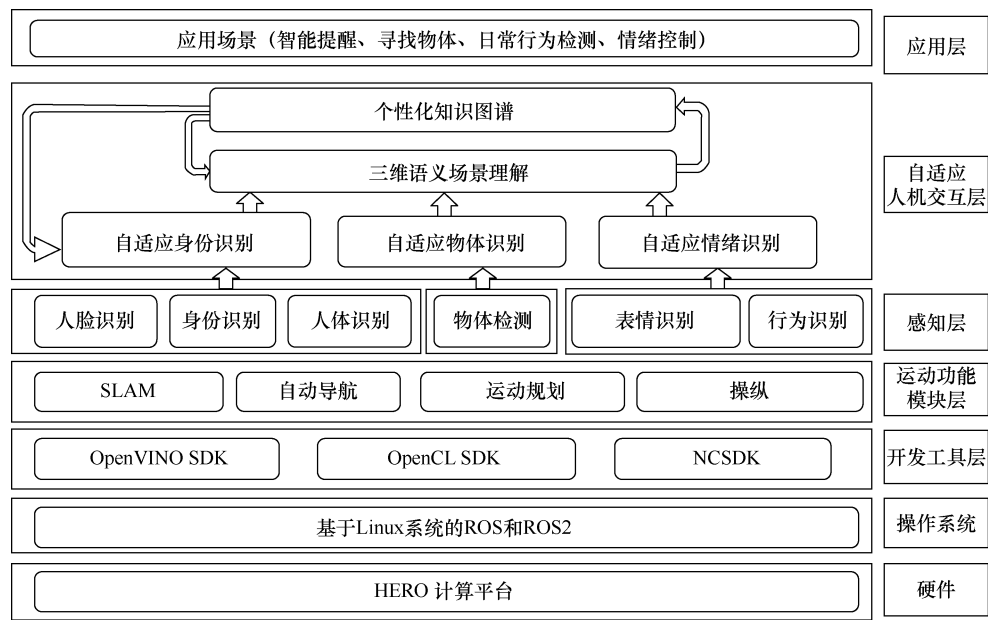


图1 开放的机器人参考平台的软件协议栈

笔者在基于 Core i7-6770HQ 的英特尔 NUC (next unit of computing) 迷你电脑平台上对上述功能模块进行工作负载评估。该计算平台基于英特尔 6 代四核 8 线程 CPU, 采用的 GPU 是核显 Iris® Pro 580, 并且配备了 Movidius 的 USB 视觉处理单元 (vision processing unit, VPU) 模块, 从而构成了一个异构的计算平台。表 1 展示了不同模块运行在不同处理器上的资源占用情况。将工作负载分配到 GPU 以及 VPU 等 AI 加速器上, 可以有效降低 CPU 的占用率, 体现了异构计算的优势。但是, 从表 1 可知, 各个模块如果同时运行, CPU 的系统占用率将会接近 400%, 系统负载非常重, 平均负载已经达到计算系统的最大算力。

虽然机器人的多种任务要求计算平台具有强大的、多样的计算能力, 但是因为机器人通常由电池供电, 这从功耗上限制了计算平台的最大计算能力, 所以在设计机器人计算平台时, 需要同时兼顾计算平台的性能和功耗。通常机器人对计算平台的功耗限制在 10~100 W。高性能的 CPU、GPU、FPGA

以及 AI 加速器虽然性能强大, 但是因为其功耗过高而无法被应用于机器人的计算系统, 所以机器人的计算平台存在计算能力的瓶颈。因此, AI 处理模块通常在机器人本体上采用轻量级的深度神经网络, 以降低对计算能力的需求, 其缺点是降低了算法的准确率, 从而降低了机器人的认知能力。

根据 PC 和手机的发展经验来看, 当市场规模足够大时, 工作负载的多样性提升得非常快, 计算系统对计算存储的要求也越来越高, 而机器人本体硬件的迭代速度和成本因素无法与之匹配, 因此机器人计算能力的扩展需要借助高速无线网络调用环境里的高性能 PC 和服务端所提供的计算资源来实现。此外, 虽然机器人需要具备处理多种任务的能力, 但是对于机器人而言, 尤其是服务机器人, 并非在全部时间内执行任务, 其多数时间并不是满负荷运行的。为最大工作负载配备计算资源也并非经济有效的解决方案, 更有效的解决方案是通过服务器来分担机器人的部分工作负载。一方面, 服务器可以为机器人提供更强大的算力, 使机器人上可以运行

表 1 不同模块运行在不同处理器上的系统占用率

处理器	视觉 SLAM	自动导航	3D 建图	图像采集	物体检测	人脸识别	人体识别	物体分割	合计
CPU	128%	34%	91.58%	49.69%	—	48.79%	—	25%	377.06%
GPU	—	—	—	—	—	100%	—	—	100%
ASIC (VPU)	—	—	—	—	1 块	—	1 块	2 块	4 块

更复杂的 AI 算法,从而使算法的精度不受到影响;另一方面,服务器可以同时为多个机器人服务,从而有效地解决机器人多任务的动态性的需求。

3 基于云原生的边缘计算系统

根据国际数据公司 (International Data Corporation, IDC) 预测^[9],未来超过 70% 的数据需要在边缘侧进行分析、处理和存储,因此边缘计算的多样性的计算架构、产品与解决方案越发重要。边缘计算是在靠近物或数据源头的网络边缘侧,融合网络、计算、存储、应用等核心能力的分布式开放平台 (架构),就近提供边缘智能服务,以满足行业数字化在敏捷联接、实时业务、数据优化、应用智能、安全与隐私保护等方面的关键需求。边缘计算的软件平台采用云原生架构与关键技术,硬件平台支持超异构计算能力,以边云协同和边缘智能为关键特征。边缘计算的总体架构如图 2 所示,形态上主要包括边缘网关、边缘云、云边缘。

- 边缘网关: 边缘网关是企业/行业数据的接

入节点,是网关设备基于云计算技术的演进,其可实现网关内资源共享。

- 边缘云: 边缘云是基于云计算技术与架构构建的边缘分布式开放平台,可提供集中管理和调度的能力,边缘云内及边缘云之间可以进行资源共享。

- 云边缘: 云边缘作为公有云的延伸,将云的部分服务或者能力扩展到边缘基础设施上。中心云和云边缘相互配合,实现全网资源共享、全网统一管控等。

在边缘网关、边缘云、云边缘的系统架构中,针对业务类型和所处边缘位置的不同,边缘计算平台的硬件选型往往也会不同。相比于硬件架构设计,边缘系统的软件架构存在一定的共同特性,比如与设备无关的微服务、资源调度和编排、容器及虚拟化技术等。以上软件技术统一了云端和边缘的服务运行环境,并提供了统一的部署和运维接口,这些技术背后的支撑是云原生软件架构在边缘的演化。边缘系统的软件架构如图 3 所示。

在云计算领域,传统 IT 软件的微服务化已经

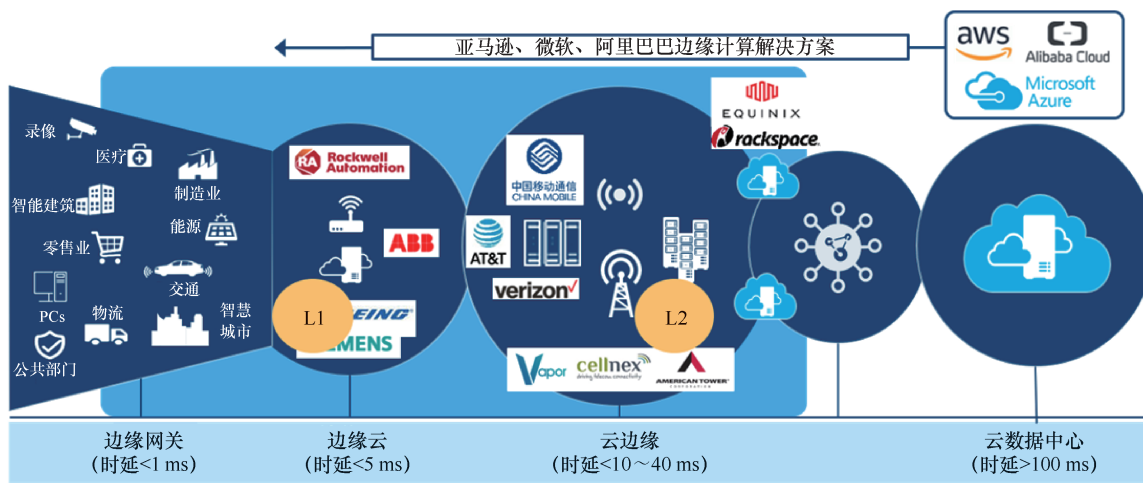


图2 边缘计算的总体架构

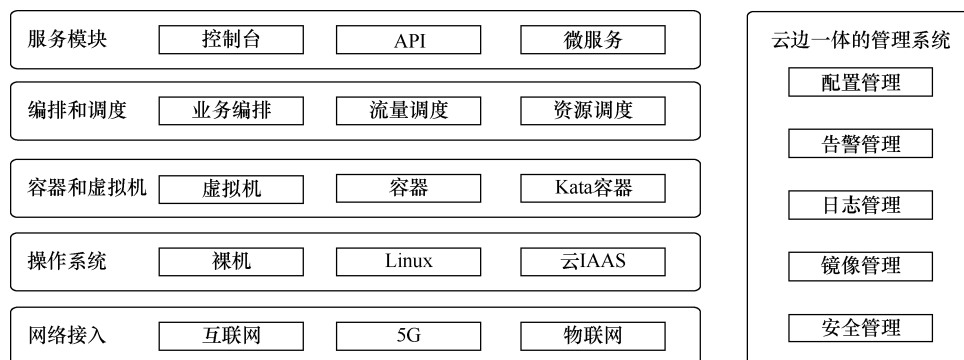


图3 边缘系统软件架构

得到了充分的演化而日趋成熟。而边缘计算是传统的工业领域的运营技术（operational technology, OT）、通信领域的通信技术（communications technology, CT）和信息技术（information technology, IT）的融合。目前仍有相当一部分 CT 和 OT 的软件是基于整体式架构并根据定制需求开发的，因此传统 CT 和 OT 的微服务化是边缘计算软件演化的关键进程。

在一些复杂的边缘计算的应用场景中可以看到，边缘计算平台并不是传统意义上的只负责数据收集和转发的网关，更重要的是，边缘计算平台需要提供智能化的计算能力，并且需要产生可操作的决策反馈来反向控制设备端。过去，这些计算只能在云端完成。现在需要从云边一体的角度，考虑通过裁剪、合并等简化手段将云端的计算框架迁移至边缘计算平台，使边缘计算平台可以运行云端训练后的智能分析算法。边缘软件架构通常被用于在单台计算机或者由少数几台计算机组成的小规模集群环境中隔离主机资源，以实现分布式计算框架的资源调度。而随着容器技术和容器编排技术逐渐成熟，通过容器技术进行资源的隔离、调度和编排将是边缘软件架构的演进方向。

4 基于边缘计算的云原生机器人系统设计

ROS^[7]起源于斯坦福大学的一个机器人项目，该项目是由开源机器人基金会（Open Source Robotics Foundation, OSRF）维护的开源项目，ROS 是机器人界被广泛接受的综合性平台。ROS 是一个适用于机器人研究和开发的开源框架，它通过提供一个基本的通信架构将原本松散的零部件耦合在一起。它在应用层提供了类似操作系统的服务，包括硬件抽象、底层设备控制、常用函数的实现、进程间消息传递以及包管理。它也提供用于获取、编译、编写跨计算机运行代码所需的工具和库函数。因此它也算是一个中间件，在基于 ROS 的应用程序之间建立起沟通的桥梁，在 ROS 中，机器人的感知、决策、控制算法可以更好地组织和运行。ROS 是一个分布式的进程（即“节点”）框架，这些进程被封装在易于分享和发布的程序包和功能包中，所有的 ROS 应用程序以 ROS 节点的形式运行。

ROS 已经有十多年的历史，最初是为独立的机器人系统而设计的，但其系统设计并不能很好地适

应云原生系统。ROS 的云原生问题如图 4 所示。首先通过 ROS CORE 启动的主节点（master）是整个 ROS 运行的核心，它的主要功能是登记节点、服务和消息主题的名称，并维护一个参数服务器。单一的主节点架构存在单点故障依赖性问题，不能很好地适用于分布式架构^[10]，目前有一些方案通过提供多主节点来解决这个问题，但存在一定的效率问题。

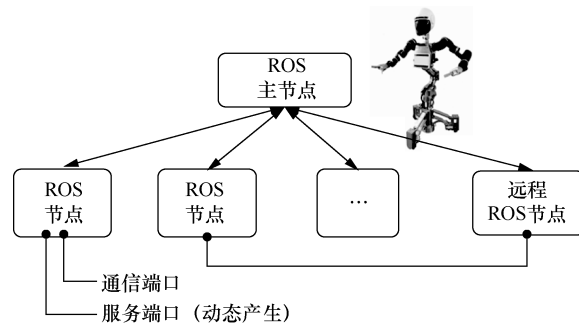


图4 ROS 的云原生问题

其次，ROS 节点在被注册至主节点时，注册了 ROS 节点的通信端口（contact endpoint），但是该节点真正的消息服务接口则需要和其他节点通信时动态产生，而动态产生的服务端点对云原生不友好，也就是说 ROS 节点在云原生系统中很难将动态产生的服务端点发布出来提供服务。这是 ROS 在云原生环境下需要解决的问题。

ROS 最初是面向机器人进行研究的，因此存在很多局限性。这些局限性对机器人整体的性能有很大影响，对 ROS 局部功能的改善很难带来整体性能的提升，因此 ROS2 应运而生。ROS2 相对于 ROS 最大的变化是改变了节点之间的通信框架。ROS 的通信系统基于 TCP/UDP，强依赖于主节点的处理，而 ROS2 的通信系统基于数据分发服务（data distribution service, DDS），由 DDS 来完成节点的发现以及消息的传输，解决了 ROS 对主节点的单点依赖问题。同时在 ROS2 内部提供了 DDS 的抽象层，通过这个抽象层，用户可以方便地选择底层 DDS 的不同实现方式。DDS 是一个专门为实时系统设计的数据发布/订阅标准，目前已成为分布式实时系统中数据发布/订阅的标准解决方案。DDS 增强了 ROS2 的实时性和安全性，很好地解决了机器人相应的需求。但是 ROS2 对云原生环境的支持也存在其特有的问题，如图 5 所示。

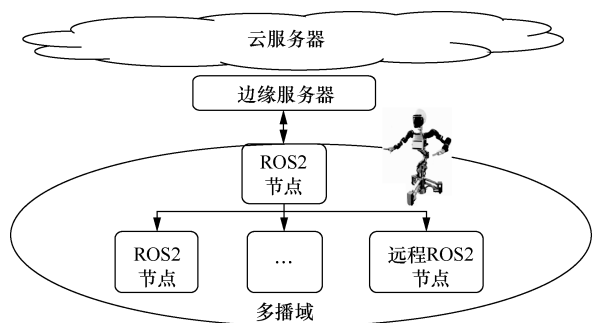


图5 ROS2的云原生问题

DDS 是为实时系统设计的数据发布/订阅的标准，理论上可以构建在 TCP、UDP 等不同的传输协议上，但是目前 DDS 的实现都是基于多播通信的，这就造成了其对多播域的强依赖。通常边缘服务器以及云服务器和客户端节点不处于同一个多播域中，因此要将 ROS2 的部分服务节点分布到边缘服务器及云服务器上，就需要解决 ROS2 跨多播域通信的问题，这也是对边缘计算的云原生系统的挑战。

总结来说，ROS 的云原生问题主要是单一 ROS 主节点和动态服务端口的问題，且 ROS 自身也存在其固有的性能问题。而 ROS2 的云原生问题主要是跨多播域通信的问题。二者都需要解决多机器人通信的问题。笔者以 ROS2 为例，讨论基于边缘计算的云原生机器人系统所要解决的问题，不仅因为 ROS2 是机器人系统未来的发展方向，还因为其相应的解决方法也适用于 ROS。云原生系统则以 Kubernetes (K8S) 为例进行分析。

K8S 是谷歌开源的一个容器编排引擎，它支持自动化部署、大规模可伸缩、应用容器化管理。谷歌将其作为初始和核心项目贡献给云原生计算基金会 (Cloud Native Computing Foundation, CNCF)，并在近年来逐渐发展出了云原生生态。K8S 作为云原生应用的基石，相当于一个云操作系统，已然成为容器编排调度的事实标准。Pod 是 K8S 能够创建和部署的最小单元，是 K8S 集群中的一个应用实例，总是被部署在集群中的同一个节点上。Pod 中包含了一个或多个容器，还包含了存储、网络等各个容器共享的资源。

4.1 桥接解决方案

实现 ROS2 对云原生支持的非常简单、直接的方式就是实现一个 ROS2 的桥接节点^[1]，借此解决 ROS2 跨多播域通信的问题。如图 6 所示，机器人本体上运行了传感器以及与运动控制相关的 ROS2 节点，而定位和 AI 感知等模块则运行在边缘服

器上。运行在边缘服务器上的 ROS2 节点通过实现容器化运行在云原生环境中，容器化的 ROS2 节点被部署在 K8S 中最基本的部署单元 Pod 之中，其可以通过云原生系统生成的网络运行在同一个多播域之中。虽然运行于边缘服务器的云原生系统的网络和机器人本体位于不同的多播域，但是可以通过位于机器人本体和边缘服务器上的 ROS2 桥接节点将它们连接在一起，完成两个多播域的消息转发，使机器人本体和边缘服务器的云原生系统构成一个虚拟的 ROS2 域。

ROS2 桥接节点之间的通信协议可以自行定制，并运行在 TCP、UDP 等传输协议之上。一对 ROS2 桥接节点连接两个物理分开的 ROS2 的多播域，可以根据 ROS2 节点在本地和边缘服务器上的分割，实现消息隔离，避免在两个多播域之间传输没有订阅关系的消息，节省通信带宽。但是桥接节点相当于应用层桥接，存在冗余的消息解析、序列化和反序列化过程，因此会影响域间通信的效率。

这种桥接方案同样可以支持非 ROS 机器人架构，通过定制化的 ROS2 桥接节点实现 ROS2 的消息与非 ROS2 的消息转换，从而实现与非 ROS 节点的通信，以支持非 ROS 机器人的实现。参考文献[12]提出的针对服务机器人的云计算框架就采用了桥接方案，将机器人系统桥接到云计算平台，但是其云计算框架并非云原生系统。

4.2 统一云原生解决方案

另外一种简单的解决方案是把机器人本体也纳入同一个云原生系统进行管理。K8S 通过容器网络接口来支持不同的容器网络的实现。其中，有的容器网络支持在云原生容器网络中构建一个跨节点的统一的多播网络域 (如 WeaveNet)，这样就通过云原生系统自身解决了机器人本体到边缘服务器跨多播域通信的问题。如图 7 所示，运行在机器人本体和边缘服务器上的 ROS2 节点分别运行在同一云原生系统的不同 Pod 之中，可以通过容器网络进行多播通信。

这种解决方案在网络层实现了不同多播域的连接，因此不存在桥接解决方案中应用层消息转换的开销。但是该解决方案因为在网络层进行连接，应用层的消息是透明的，无法隔离机器人本体和边缘服务器之间无订阅关系的消息，并不能有效地节省网络带宽。此外，虽然云原生系统支持将不同的微服务部署到特定的主机上，但是由于机器人本体

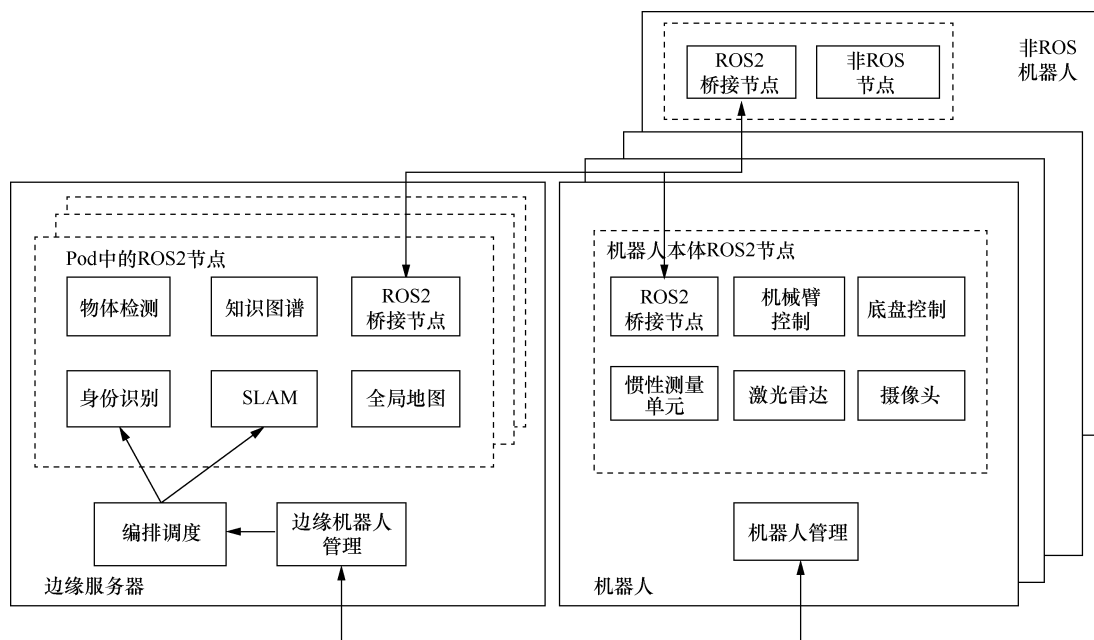


图6 桥接解决方案

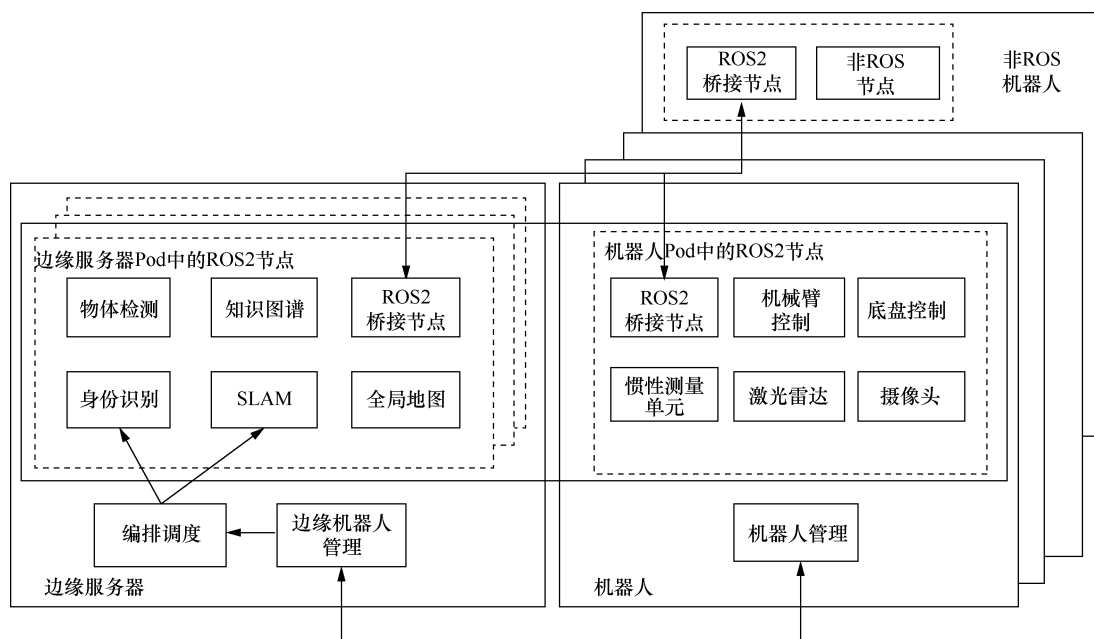


图7 统一云原生解决方案

存在大量的传感器和控制器，容器化的 ROS2 节点需要直接对外部设备进行访问，这对云原生系统的要求比较高，需要在云原生系统中支持多种外部设备，因此这种解决方案需要很大的工作量。

4.3 覆盖（overlay）网络解决方案

在统一云原生解决方案中，云原生网络实现了一个虚拟的网络层，使不同主机上的容器可以互相通信。同样地，也可以在机器人本体与边缘服务器的云原生系统的容器网络之间构建一个虚拟的网

络层，使机器人本体在不加入边缘服务器的云原生系统的情况下，工作在同一个叠加的多播域中，从而形成一个覆盖网络的解决方案。

这种解决方案在网络通信效率上和统一云原生解决方案是相同的，但是不需要云原生系统支持机器人本体的各种外部设备。但是实现这种解决方案需要对云原生系统的网络接口进行改进，以达到在边缘服务器的云原生系统中接入机器人本体的目的。

4.4 非 ROS2 边缘服务解决方案

ROS2 的核心是提供了一种节点发现以及消息发布/订阅的机制, ROS2 本身对云原生并不友好。在云原生系统中, 同样有类似的技术实现节点发现和消息发布/订阅的功能, 比如谷歌的发布/订阅(pub-sub)接口, 这些技术都是云原生支持的, 因此可以利用这些技术在边缘服务器上实现非 ROS2 的机器人微服务。如图 8 所示, 边缘服务器云原生系统中的微服务节点是非 ROS2 的服务节点, 与桥接解决方案相似, 可以通过 ROS2 桥接节点将边缘服务器云原生系统与机器人本体的 ROS2 系统连接起来, 共同实现一个完整的机器人系统。参考文献[13]提出的云机器人平台采用了非 ROS 的通信协议 WebSocket, 基于 WebSocket 协议开发机器人服务十分简单, 但是对机器人系统的实时性要求存在一定的局限性。

利用非 ROS2 的云原生的数据发布/订阅技术可以更好地在边缘服务器中进行编排和调度, 最大

化利用云原生系统的优势, 从而优化机器人系统的性能, 但是因为采用了非 ROS2 的通信技术, 边缘服务器的微服务很难利用到 ROS2 的生态系统, 这为实现这种解决方案带来了工程上的挑战。

以上 4 种解决方案各有优劣, 在实际系统中采用哪种解决方案, 取决于实际的机器人应用场景, 包括机器人的应用特点, 各个模块所采用的技术方案对延迟、带宽、性能以及成本的要求。表 2 对这 4 种方案进行了对比。

本文主要讨论了基于边缘计算的云原生机器人系统。边缘服务器可以被看作云计算在边缘的延伸, 云原生系统本身来自云计算, 因此本文讨论的方案同样也可以被应用到云计算系统中, 构成一个云-边-端融合的云原生机器人系统。

基于边缘计算的云原生机器人系统不仅为终端机器人系统提供了边缘服务器强大的计算资源, 同时也更有效地服务于多机器人系统, 而且, 只有在多机器人的系统中才能更好地体现云原生机器

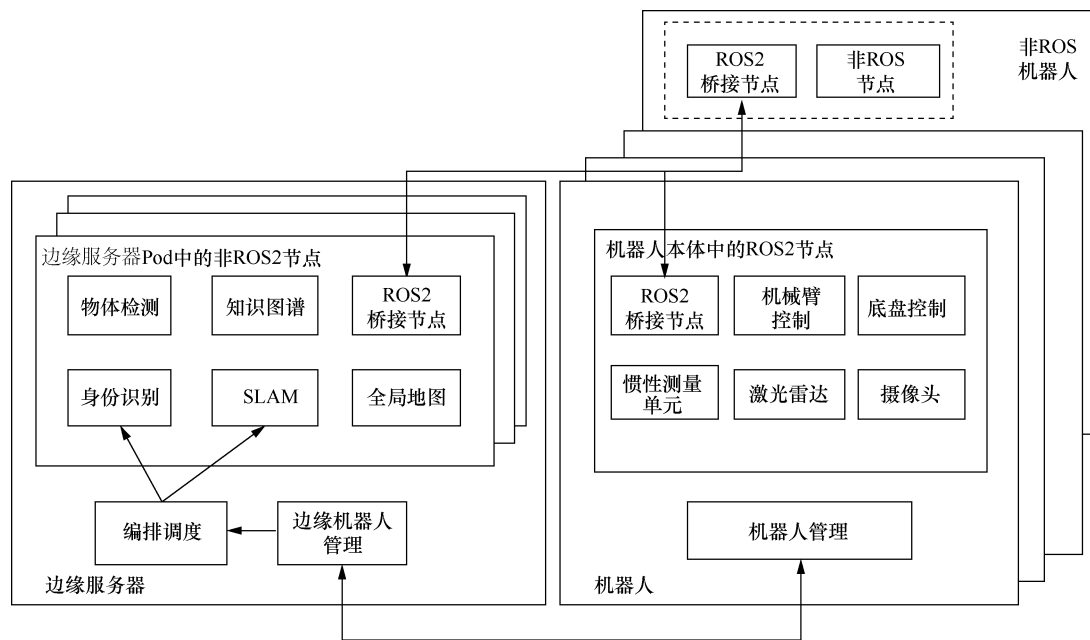


图 8 非 ROS2 边缘服务解决方案

表 2

ROS 云原生解决方案比较

方案	通信层级	通信开销	带宽优化	开发成本	系统性能	部署难度
桥接解决方案	应用层	高	易	低	低	高
统一云原生解决方案	网络层	低	难	高	高	高
覆盖网络解决方案	网络层	低	难	高	高	低
非 ROS2 边缘服务解决方案	应用层	高	易	高	高	低

人系统的优势。本文提出的4种解决方案如何更好地服务于多机器人系统，还需要进行更深入的研究和探索。

5 结束语

机器人技术是多学科交叉的科学工程，涉及机械、电子、计算机、通信、人工智能和传感器，甚至纳米科技和材料技术等。智能机器人可谓人工智能应用“皇冠上的明珠”。

边缘计算是最近提出的新概念，其推广和落地非常依赖于具体的应用，尤其适用于对实时性要求较高的场景，机器人应用正是非常合适的应用场景。本文提出的基于边缘计算的云原生机器人系统是实现可产业化应用的云-边-端融合的机器人系统的核心。基于边缘计算的云原生机器人系统可以使机器人本体充分利用边缘的实时计算能力和云计算的无限处理能力，从而增强机器人本体自主移动的能力，大大提升机器人的感知和推理能力。云边端一体化使机器人本体的能力具有很大的弹性空间，进而降低了机器人对本体硬件能力的依赖和成本，使机器人的大规模部署成为可能。

参考文献：

- [1] 中国电子学会. 中国机器人产业发展报告 2019[R]. 2019. Chinese Institute of Electronics. 2019 China robotic industry development report[R]. 2019.
- [2] 国际数据公司. 人工智能时代的机器人 3.0 新生态[Z]. 2017. IDC. Robot 3.0: new ecology in the age of artificial intelligence[Z]. 2017.
- [3] Intel Labs China, CloudMinds Technology Inc., SIASUN Robot & Automation Co., Ltd. Ecovacs Robotics. Robot 4.0 Whitepaper[Z]. 2019.
- [4] WANG Z G, WANG H T, SHE Q, et al. Preliminary analysis on key technology of robot 4.0[J]. Journal of Computer Research and Development, 2020, 57(9): 1854-1863.
- [5] Intel Labs China. Adaptive huma-robot interaction (AHRI) whitepaper[Z]. 2019.
- [6] Intel Labs China. Adaptive human-robot interaction (AHRI) SDK 2.1[Z]. 2020.
- [7] QUIGLEY M, CONLEY K, GERKEY B P, et al. ROS: an open-source robot operating system[C]// ICRA Workshop on Open Source Software. [S.L.:s.n.], 2009.
- [8] SHI X S, CAO L, WANG D W, et al. HERO: accelerating autonomous robotic tasks with FPGA[C]// 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE Press, 2018.
- [9] IDC. IDC futurescape: worldwide IoT 2020 predictions[Z]. 2019.
- [10] AMY M, LAUER M, FABRE J C. Adaptive fault tolerance on ROS: a component-based approach[C]// ROSCon2016. [S.L.:s.n.], 2016.
- [11] CRICK C, JAY G, OSENTOSKI S, et al. ROS and rosbridge: roboticists out of the loop[C]// The 7th Annual ACM/IEEE

International Conference on Human-Robot Interaction. New York: ACM Press, 2012: 493-494.

- [12] ARUMUGAM R, ENTI V R, BINGBING L, et al. DAvinCi: a cloud computing framework for service robots[C]// IEEE International Conference on Robotics & Automation. Piscataway: IEEE Press, 2010: 3084-3089.
- [13] MOHANARAJAH G, HUNZIKER D, D'ANDREA R, et al. Rapyuta: a cloud robotics platform[J]. IEEE Transactions on Automation Science & Engineering, 2015, 12(2): 481-493.

[作者简介]



王大伟（1974—），男，博士，英特尔（中国）研究中心有限公司英特尔中国研究院系统架构师，主要研究方向为机器人的软件系统架构、机器人系统优化以及云边端融合的机器人系统。



王卓（1972—），男，英特尔（中国）研究中心有限公司英特尔中国研究院系统架构师，主要研究方向为机器人的硬件系统架构、异构计算平台、传感器融合以及硬件加速器。



王鹏（1979—），女，博士，英特尔（中国）研究中心有限公司英特尔中国研究院技术规划与战略合作总监，主要研究方向为服务机器人、视觉计算、人机交互等。曾获得英特尔全球品牌大使、英特尔技术最高奖、英特尔中国最佳员工以及国际消费电子展最佳创新奖、最佳无线应用奖、最佳技术奖。发表国内外学术论文30余篇，申请国际专利10余项。



王志刚（1978—），男，博士，英特尔（中国）研究中心有限公司英特尔中国研究院研究员，中国计算机学会会员，主要研究方向为机器人的时空智能、多层次AI认知系统及机器人系统优化。



王海涛（1978—），男，英特尔亚太研发有限公司软件架构师，中国计算机学会会员，主要研究方向为边缘计算软硬件系统设计、云原生软件架构、时间敏感网络及机器人系统优化。