

设备无感知边缘模型协同推理

王 帅 李博睿 王维龙 等
东南大学

引言

近年来,人工智能(AI)技术持续进步,特别是针对文本、图像、语音、视频等多种模态的生成式模型迅速普及,其应用逐渐渗透至人们的日常生活中,改变了人类获取甚至生成信息的方式。模型部署框架是一种软件工具集,能帮助算法开发人员将所训练的算法模型部署到生产环境,转化为可调用的服务,并确保模型在各种硬件加速器与用户请求下均能够高效、稳定运行。现有面向云服务器的模型部署框架较为成熟,以开源框架为例,有谷歌提出的 TensorFlow Serving 和基于容器的云原生模型推理框架 KServe。

除了云端模型服务之外,考虑到数据的隐私性、服务的实时性和模型的个性化等特性,越来越多的 AI 模型和应用开始针对边缘侧进行有针对性的设计和优化。例如,由 Meta 公司开发并开源的 Llama 模型家族,在其 2024 年的升级中,除了传统的 110 亿参数和 900 亿参数的“大”模型,还新增了面向边缘设备部署、仅包含 10 亿和 30 亿参数的轻量级模型 Llama 3.2。随着边缘侧智能模型和应用的不断涌现,边缘模型部署框架作为连接上层模型算法与底层边缘计算软硬件(AI 模型算子、AI 加速硬件等)的桥梁,被认为是一个重要发展方向。然而,设计一个面向边缘的模型部署框架并非易事,如何面向边缘计算场景中独特的分布式设备协同、边缘设备能力异构以及边缘部署场景复杂等特点构建一个灵活、高效的模型部署框架,是亟待解决的问题。

针对上述问题,基于设备无感知计算(deviceless computing)的模型部署范式应运而生。顾名思义,设备无感知计算的核心是使开发部署人员感受不到边缘侧分布式多设备的存在,后续的推理任务在各个设备的部署与调度,以及面对突发状况的推理迁移均会自动完成。近两年来,设备无感知模型部署受到国内外工业界与学术界的广泛关注。在工业界,世界最大的云服务提供商亚马逊 AWS 推出了 SageMaker Edge 实现边缘侧设备无感知 AI 部署,国内的阿里云、腾讯云也都推出了各自的产品。在学术界,IEEE 会士、维也纳技术大学(TU Wien)教授沙赫兰·杜斯塔(Schahram Dustdar)多次发表论文展望设备无感知计算与 AI 部署的未来发展,东南大学、浙江大学、天津大学、华中科技大学、中国地质大学(武汉)、加州大学河滨分校、佐治亚理工学院、帝国理工学院等国内外知名高校的研究团队也在相关方向发表了高水平研究成果。

本文在介绍设备无感知边缘模型协同推理的发展与整体框架的基础上,从基础协同环境、推理实例调度与推理场景优化三个方面探讨边缘模型协同推理的关键技术,并在此基础上对未来技术发展趋势进行展望。

从服务器无感知到边缘设备无感知计算:发展、挑战与框架

从历史沿革来看,边缘设备无感知计算是脱离于服务器无感知计算(serverless computing)、面向

边缘场景的新型计算范式。因此，在介绍边缘无感知计算之前，本文首先对服务器无感知计算的发展历史进行回顾，进而借助边缘计算模型部署场景的三大特点引出边缘设备无感知计算的三层架构。

根据 CCF 计算机术语审定委员会的定义^[1]，服务器无感知计算是“以函数为中心”的计算范式，开发者只须专注于编写云函数实现应用逻辑，云资源管理和环境配置完全由云计算系统软件负责，对开发者透明。在此基础上，服务器无感知框架通过自动扩缩容策略，根据应用流量动态增减函数实例，在没有请求时完全关闭函数实例，提高整体资源利用率。ACM 图灵奖获得者戴维·帕特森（David A. Patterson）在其论著中称服务器无感知计算将成为“云时代的默认计算范式”（the default computing paradigm of the cloud era）^[2]。使用服务器无感知框架部署 AI 模型，可极大降低算法开发者的管理负担，正成为一个学术研究焦点。在国际知名的 ACM 欧洲计算机系统会议（ACM EuroSys）上，关于服务器无感知计算的研讨会“SErverless Systems, Applications and MEthodologies”（SESAME）已连续举办三届，并且最新一届研讨会的主题就是“服务器无感知如何更好地服务包括生成式模型在内的 AI 算法部署”。

尽管服务器无感知计算是为云端应用部署设计的，但其愿景与架构优势在边缘侧同样非常重要。例如其自动扩缩容特性非常适合请求数量与频次多变但计算资源有限的边缘计算场景；其基于沙箱隔离技术的函数化部署思想，面对复杂的边缘侧部署软硬件环境具有重要的实际意义。

然而，相较于云计算，边缘计算基础设施具有特殊性，若想将服务器无感知计算应用于边缘侧模型部署并非易事。具体而言，需要解决以下三大挑战：

1. 设备分布协同困难。与在数据中心中实现大规模 AI 部署不

同，边缘 AI 部署场景通常是以云－边协同或边－边协同的分布式方式，并且设备间往往通过网络互联。如何在分布式环境下，使开发者能够关注模型本身而非繁杂的分布式系统细节，赋予模型快速部署与服务的能力，是边缘模型部署框架需要解决的问题之一。

2. 边缘推理能力异构。从总体来看，边缘设备的算力远弱于服务器。并且由于边缘设备之间部署位置、电源供给等方面的不同，边缘设备的硬件配置和算力各异。针对设备推理能力异构问题，如何通过高效的资源调度利用好有限的资源，使其能够服务更多推理请求显得尤为重要。

3. 模型部署场景复杂。除了基本的模型推理场景之外，由于边缘侧距离数据源更近、数据安全考虑更复杂，衍生出了持续学习、联邦学习等不同模型部署场景。如何在边缘侧模型部署系统中更好地支持多样部署场景，也是需要解决的关键问题之一。

面对上述三大挑战，本文将设备无感知边缘模型协同推理框架分为协同环境层、推理调度层、场景优化层三个层次，如图 1 所示。协同环境层主要提供 AI 模型运行的沙箱环境，并在此基础上实现设备无感知的协同与推理迁移。推理调度层需要考虑边缘设备的算力调度，并在此基础上实现设备无感知计算模型的关键调度策略——自动扩缩容。场景优化层主要针对不同的特定模型部署场景（如持

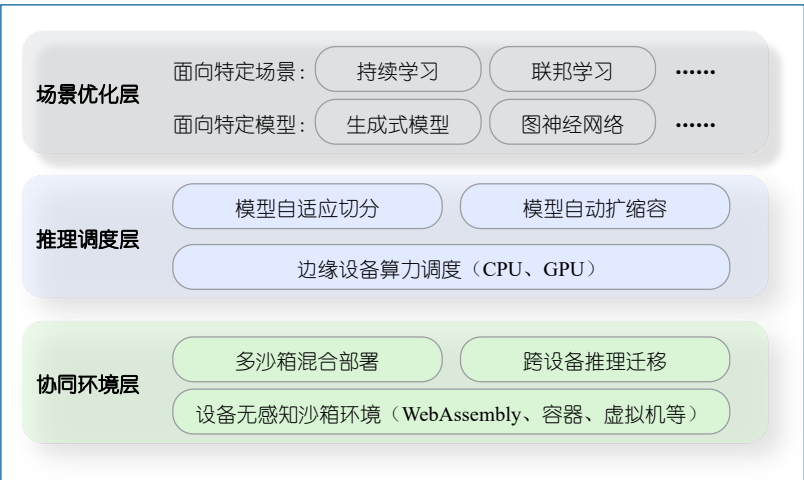


图1 设备无感知计算层次架构

续学习、联邦学习)与特定AI模型架构(如图神经网络与生成式模型)设计对应的部署系统。

设备无感知协同环境层

设备无感知沙箱环境

在设备无感知部署方式中,每个AI模型的执行表现为单个运行的函数实例。为充分利用边缘设备,往往采用多租户部署的方式,即来自多个用户或应用的函数实例都部署在同一个边缘设备上。因此,需要将函数实例的运行封装在隔离的沙箱环境中,而所采用的沙箱环境对最终应用的性能有重要影响。

随着隔离技术的发展,各种面向服务器无感知或边缘无感知的沙箱环境层出不穷,如虚拟机^[3]、容器^[4]、WebAssembly^[5]等。虚拟机能够完整地模拟硬件运行,允许在单个物理服务器上运行多个独立的操作系统实例,其隔离性和安全性较强。容器是一种架构在操作系统之上的隔离技术,它可以共享宿主操作系统内核,通过Linux进程组隔离(cgroups)机制实现对应用的隔离。容器相比虚拟机开销更小,启动速度更快,目前是云原生计算中最通用的隔离技术。WebAssembly是一种新的二进制可执行文件格式,其体积小、加载速度快、性能接近原生运行,支持作为C、C++、Rust等多种高级语言的编译目标。此外,WebAssembly还具

有跨平台性,可在不同硬件架构(如x86、ARM、RISC-V)上执行,为在异构的边缘设备间实现无缝计算迁移与协同提供了机会。这些虚拟化技术的发展和成熟,为设备无感知的边缘模型协同推理提供了技术支撑。

多沙箱混合部署与跨设备推理迁移

虽然虚拟机、容器与WebAssembly等均为设备无感知的工作提供了稳健的平台支撑,然而不同的虚拟化技术具备不同的性能优势。例如虚拟机可以提供完整安全的计算环境,然而其运行时开销较大,管理复杂性较高。容器的效率虽然比虚拟机高,但是在执行AI任务时需要加载大量的基础环境,冷启动时间较长。容器和虚拟机的冷启动时间比WebAssembly长上百倍^[6],但是基于WebAssembly实现AI模型推理必须依赖特定的接口,其在大批量模型推理工作方面的性能较容器还有差距。

因此,研究人员提出根据所部署的应用和算法在多种沙箱技术中动态地选择一种,实现多沙箱混合部署,从而有效补全不同环境的计算特性短板。如Mark^[7]和Sponge^[8]将容器与虚拟机混合部署,利用容器实现动态可扩展性,利用虚拟机保持稳定的推理性能。然而虚拟机的执行开销较大,不适合边缘计算环境。因此,课题组提出将容器与WebAssembly混合部署,实现AI算法的高效部署与服务。如图2所示,通过对AI模型的权重与设备无感知计算函数的联合分析,选择不同的沙箱部署环境。此外,由于

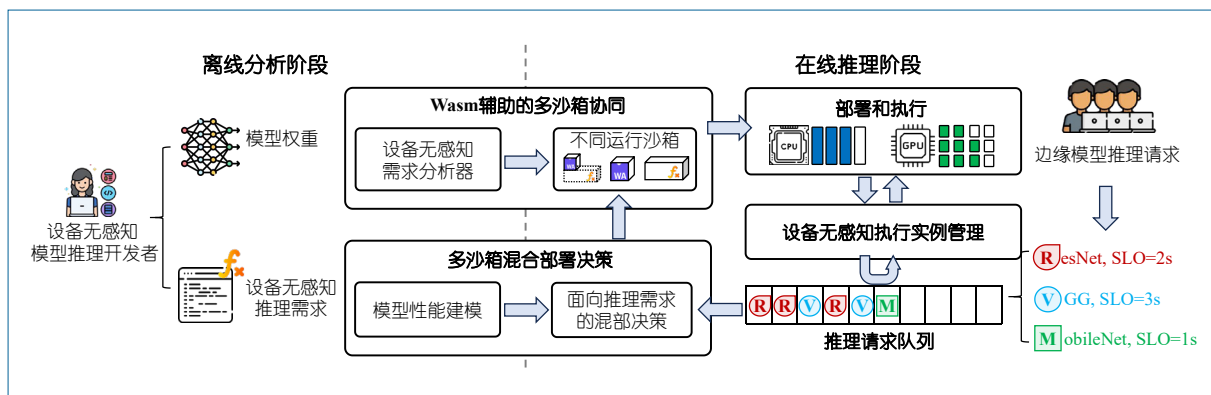


图2 面向边缘模型推理的WebAssembly与容器多沙箱混合部署

WebAssembly 冷启动性能较好,在选择容器的时候,使用 WebAssembly 帮助容器在冷启动期间提前完成部分模型推理准备工作,例如文本向量化、图片灰度化/二值化等。在此基础上,根据每个边缘 AI 推理请求的服务等级目标 (Service-Level Objective, SLO) 对混合部署实例进行管理与调度。

由于设备无感知计算范式需要管理分布式设备间的部署,而无须管理底层部署细节。因此,实现设备间计算函数迁移尤为重要。对于基于容器技术的函数,研究者提出使用 Linux 的用户态检查点 (Checkpoint/Restore In User-space, CRIU) 机制在容器任意运行状态实现跨设备迁移^[9]。针对 WebAssembly 函数实例,笔者所在课题组提出轻量级执行流截断与紧凑内存快照等技术^[5],实现跨异构设备的高效执行上下文打包与执行迁移。

设备无感知推理调度层

推理调度层主要包括边缘设备算力调度、模型自适应切分与模型自动扩缩容,为设备无感知的模型推理提供高效且精确的资源调度。

在设备算力调度方面,已有面向云端服务器无感知模型推理工作通常将系统设计与 AI 本身的计算特性相结合,中山大学教授周知团队提出的 λ DNN^[10] 通过动态调整分配给各个函数的资源来满足 AI 模型时延要求;休斯顿大学教授颜枫团队提出的 BATCH^[11] 通过动态调整 AI 推理的批处理大小 (batch size),在推理的尾部延迟和推理成本最小化的情况下实现高效的设备无感知 AI 模型推理。

在此基础上,设备无感知计算的一大特点是模型自动扩缩容。天津大学教授李克秋团队提出的 INFless^[12] 支持成批处理原

生与云端异构计算资源调度,通过协调管理批处理的排队时间和冷启动模式实现高吞吐、低延迟的 AI 模型推理。华中科技大学教授刘方明团队提出的 AsyFunc^[13] 采用非对称的服务器无感知推理的扩缩容策略,根据 AI 模型的计算、存储资源使用情况进行推理调度。

上述已有工作主要针对云计算场景的无服务器推理进行优化,然而,与云端丰富的计算资源不同,边缘推理设备的资源有限。例如,推理 6-bit 量化 Llama2-7B 模型需要约 14 GB 内存,而由于功耗与体积的限制,配备 16 GB 内存的边缘设备已较为少见。因此,设备无感知推理调度时无法像调度云端函数一样,通过直接扩容整个实例来满足突发的推理请求。

针对此问题,笔者所在课题组通过大量的初步实验发现,AI 模型内部不平衡的计算复杂性会导致推理瓶颈^[14],通过整体扩缩容的方式给整个 AI 模型分配等比例的更多资源,会导致推理瓶颈处资源不足、非瓶颈处资源过剩,从而阻碍整体推理性能的提升。因此,课题组提出推理瓶颈感知的设备无感知推理选择性扩缩容方法,如图 3 所示。在部署推理之前的离线阶段,课题组提出针对瓶颈感知的模型自适应切分方法,它能够在模型执行前根据模型中算子间的拓扑关系与计算特征识别出潜在的性能瓶颈,并将这些瓶颈部分独立切分,为在线选择性扩缩容做好准备。在线推理阶段,通过观测模型的调用情况与每个潜在性能瓶颈的表现,选择影响当前整体推理效率的推理性能瓶颈,并选择性地进行在线扩缩容,以此实现瓶颈感知的边缘推理。另

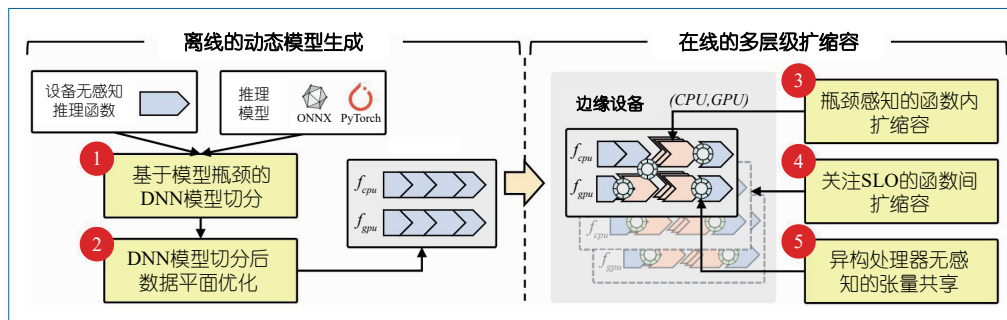


图3 推理瓶颈感知的设备无感知推理选择性扩缩容

外,由于功耗和尺寸的限制,边缘设备的硬件设计有一个显著特性,即在单个片上系统(SoC)上集成多个推理加速硬件,如NVIDIA Jetson系列在同一块SoC上集成了CPU和GPU,并利用统一内存实现CPU与GPU之间的直接数据共享;Kunpeng Pro集成了CPU和AI加速单元。针对此特性,在选择性扩缩容中进一步支持高效的异构处理器无感的张量共享机制,以充分发挥边缘设备上异构计算加速硬件的能力,使处理器间的数据传输开销最小化。

设备无感知场景优化层

针对边缘侧模型部署场景复杂的问题,场景优化层负责对上层的特定模型或特定场景进行优化与支持。这些工作通常面向此模型架构或场景的首要特点,例如大语言模型的自回归推理特性或联邦学习中大量的多方通信等,在通用模型部署框架基础上进行针对性优化。

当前已有一些针对云端面向图神经网络、生成式AI模型等特定模型进行优化的服务器无感知计算系统,这些工作为未来实现边缘侧设备无感知模型优化提供了有益借鉴。例如面对图神经网络模型,北京大学教授刘譞哲团队提出的ElasticFlow^[15]关注在图神经网络部署时使用GPU资源的边际效用递减效应,有针对性地进行计算资源分配以满足延迟要求。对于生成式模型,英国爱丁堡大学的研究团队提出了ServerlessLLM^[16],通过优化承载大语言模型(LLM)的服务器无感知函数在服务器集群间的部署和迁移,实现高效的LLM推理。

除了针对特定模型的优化,场景优化层还需要支持其他多种边缘模型推理场景。在边缘环境中,联邦学习和持续学习是两类最常见的推理场景。

由于边缘计算设备距离数据源头更近,通常被认为是比云计算更安全的计算方式。然而由于现实业务场景中多样的数据安全策略,使用联邦学习在多个边缘服务器间进行协同学习的应用较为广泛。针对此场景, λ -FL将联邦学习的每个部分封装在单

一的函数中,并通过调度函数的执行实现高效的联邦学习。

另一个在边缘计算中常见的推理场景是持续学习。它既需要在推理数据漂移(即推理数据与训练数据分布不同)时使用新数据进行再训练,并基于之前任务积累的经验,在新的数据上很好地完成任务;又需要避免遗忘问题,对曾经训练过的任务依旧保持很高的精度。在云端使用服务器无感知架构支持持续学习已有开展一系列研究工作,大部分通过针对性调节函数资源分配等参数,实现持续学习效率最大化。

然而,这些框架并未将持续学习所提升的准确度与为持续学习函数实例分配的资源之间的关联性考虑进来。例如应用中的某模型精度已达到上限,再进一步进行持续学习也无法改善其精度时,应减少或停止其设备无感知函数的资源分配,以保证资源使用效率。另外,由于当前的AI应用通常使用多个AI模型级联构成有向无环图(Directed Acyclic Graph, DAG)的方式构建,应用中多个AI模型之间的数据漂移关联关系也需要考虑。例如,若1与2两个AI模型级联,在发生数据漂移时,1、2的精度均有明显下降,可能1、2都需要重新训练,也可能是由于1的精度下降导致2的精度下降,仅重新训练1即可提高两者精度。

针对上述两个问题,课题组针对视频分析应用,提出了面向多AI模型的边缘设备无感知持续学习系统,如图4所示。具体地,系统通过分析多个AI模型之间的数据漂移相关性,建立模型互相关矩阵。依据此矩阵,在分配设备无感知函数的资源时,根据各个模型对精度恢复的重要性进行重训练优先排序,最终形成资源最优的重训练计划。在此基础上,课题组还创新性地提出了利用设备无感知计算函数保活期的空余资源进行持续学习,在不增加训练成本的情况下提高整体资源效率。

展望

本文探讨了大小模型端云协同场景中基于设备

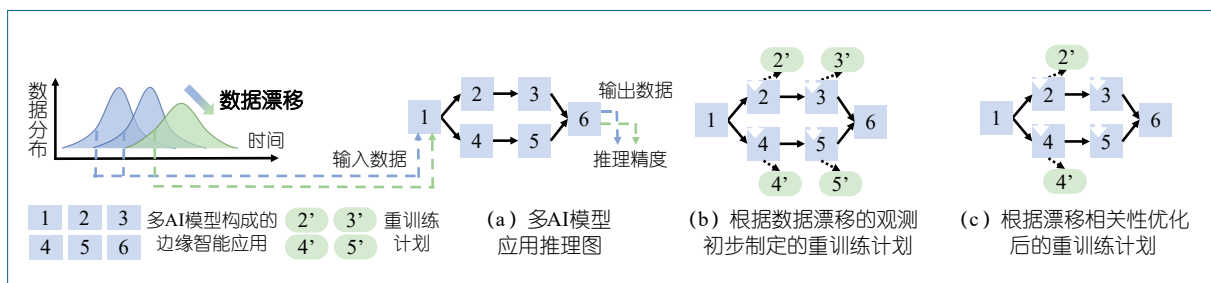


图4 面向多AI模型的设备无感知持续学习

无感知计算思想进行 AI 模型跨设备协同部署范式，从协同环境、推理调度、场景优化三个层次将相关工作进行了系统性梳理。展望未来，尽管设备无感知边缘模型协同推理技术研究已取得长足进步，但其在基础框架的开源与标准化、资源管控的跨层优化等方面仍有较大的探索空间。

1. 设备无感知框架的开源与标准化。云端的服务器无感知技术已有一系列开源框架，如 Knative、OpenFaaS、OpenLambda 等。然而，面对边缘设备无感知计算范式，目前仍未有一个具有较大影响力的开源框架，限制了其广泛应用。可喜的是，边缘设备无感知计算框架的一些重要组成部分已有一些较为成熟的开源项目支撑，例如针对边缘函数运行环境管理的 KubeEdge、OpenYurt、K3s 等。如何在这些开源项目的基础上，面向边缘场景的特点设计边缘设备无感知计算框架仍待进一步探索。

2. 设备无感知资源管控的跨层优化。针对边缘设备无感知函数的调度、迁移已有一系列优秀的研究工作。然而，现有的探索主要集中在边缘平台的调度，例如函数保活时间、部署方式、资源分配等，没有与底层操作系统调度与网络资源控制协同起来，难以达到全局最优配置。因此，如何针对设备无感知计算特点，探索实现应用－平台－系统－硬件的联合跨层优化，有助于边缘模型推理效率的大幅度提升。

3. 未来代理式人工智能系统的部署。随着大语言模型能力的不断增强，基于它的代理式人工智能（agentic AI）被著名咨询机构 Gartner 列为 2025 年十大战略技术趋势之一，英特尔、高通、微软等边缘软硬件企业均在此领域部署研究。类似的，加州

大学伯克利分校的人工智能研究中心也提出了复合人工智能系统（compound AI system）的概念，即未来的人工智能应用将不单单是一系列模型的简单组合，而是具有众多组件的复杂智能系统。面向未来代理式人工智能发展的大趋势，如何借助设备无感知范式，让 AI 智能代理在边缘侧拥有高效运行的大规模落地应用，也是一个极具未来前景的发展方向。在此方面，Linux 开源基金会已经发布了可扩展、高可用的边缘智能代理运行环境 SPEAR 项目^[17]，希望以此为契机推动设备无感知范式的进一步发展。



王 帅

CCF 高级会员、CCCF 专题编委。东南大学青年首席教授、计算机工程系主任、智慧物联网研究中心主任。主要研究方向为智能原生物联网、大数据分析。
shuaiwang@seu.edu.cn



李博睿

CCF 专业会员。东南大学讲师、至善青年学者。主要研究方向为智能物联网与边缘计算系统、设备无感知计算。libr@seu.edu.cn



王维龙

CCF 学生会员。东南大学博士研究生。主要研究方向为设备无感知计算。
wang_wl@seu.edu.cn

其他作者：刘天恩

参考文献

- [1] “服务器无感知计算”术语发布[EB/OL]. [2025-01-25]. https://www.ccf.org.cn/Media_list/gzwyh/jsjsysdwyh/2023-03-17/789780.shtml.
- [2] JONAS E, SCHLEIER-SMITH J, SREEKANTI V, 等. Cloud Programming Simplified: A Berkeley View on Serverless Computing: UCB/EECS-2019-3[R/OL]. EECS Department, University of California, Berkeley, 2019. <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-3.html>.
- [3] AGACHE A, BROOKER M, FLORESCU A, 等. Firecracker: Lightweight Virtualization for Serverless Applications[C]//Proc. of USENIX NSDI. 2020: 419-434.
- [4] CARAZA-HARTER T, SWIFT M M. Blending containers and virtual machines: a study of firecracker and gVisor[C]//Proc. of ACM VEE. 2020: 101-113.
- [5] LI B, DONG W, YI G. WiProg: A WebAssembly-based Approach to Integrated IoT Programming[C]//Proc. of IEEE INFOCOM. 2021: 1-10.
- [6] LYU X, CHERKASOVA L, AITKEN R, 等. Towards efficient processing of latency-sensitive serverless dags at the edge[C]//Proc. of ACM EdgeSys. 2022: 49-54.
- [7] ZHANG C, YU M, WANG W, 等. Mark: Exploiting Cloud Services for Cost-Effective, SLO-Aware Machine Learning Inference Serving.[C]//Proc. of USENIX ATC. 2019: 1049-1062.
- [8] SONG W W, UM T, ELNIKETY S, 等. Sponge: Fast Reactive Scaling for Stream Processing with Serverless Frameworks[C/OL]//Proc. of USENIX ATC. 2023: 301-314[2024-01-22]. <https://www.usenix.org/conference/atc23/presentation/song>.
- [9] BARBALACE A, KARAOUI M L, WANG W, 等. Edge computing: the case for heterogeneous-ISA container migration[C]//Proc. of ACM VEE. 2020: 73-87.
- [10] XU F, QIN Y, CHEN L, 等. λ dnn: Achieving predictable distributed DNN training with serverless architectures[J]. IEEE Transactions on Computers, 2021, 71(2): 450-463.
- [11] ALI A, PINCIROLI R, YAN F, 等. Batch: Machine learning inference serving on serverless platforms with adaptive batching[C]//Proc. of IEEE SC. IEEE, 2020: 1-15.
- [12] YANG Y, ZHAO L, LI Y, 等. INFless: a native serverless system for low-latency, high-throughput inference[C]//Proc. of ACM ASPLOS. 2022: 768-781.
- [13] PEI Q, YUAN Y, HU H, 等. AsyFunc: A High-Performance and Resource-Efficient Serverless Inference System via Asymmetric Functions[C/OL]//Proc. of ACM SoCC. New York, NY, USA: Association for Computing Machinery, 2023: 324-340[2024-01-22]. <https://dl.acm.org/doi/10.1145/3620678.3624664>.
- [14] BORUI LI, TIANGE XIA, SHUAI WANG, 等. InfScaler: Enabling Efficient ML Inference Serving on Multi-Accelerator Edge Devices via Asymmetric Auto-Scaling[C]//Proc. of IEEE/ACM DAC. 2025.
- [15] GU D, ZHAO Y, ZHONG Y, 等. ElasticFlow: An Elastic Serverless Training Platform for Distributed Deep Learning[C]//Proc. of ACM ASPLOS. 2023: 266-280.
- [16] FU Y, XUE L, HUANG Y, 等. ServerlessLLM: Locality-Enhanced Serverless Inference for Large Language Models[C/OL]//Proc. of USENIX OSDI. USENIX Association, 2024[2024-06-13]. <http://arxiv.org/abs/2401.14351>.
- [17] LINUX FOUNDATION EDGE. Scalable and Performant Edge Agent Runtime[CP/OL]. InfiniEdge AI, 2025[2025-02-23]. <https://github.com/lfedgeai/spear>.