arXiv:2305.02697v1 [cs.DC] 4 May 2023

DECICE：设备-边缘-云智能协作框架

Julian Kunkel[1]，Christian Boehme[2]，Jonathan Decker[1]，Fabrizio Magugliani[3]，Dirk Pleiter[4]，Bastian Koller[5]，Karthee Sivalingam[6]，Sabri Pllana[7]，Alexander Nikolov[8]，Mujdat Soyturk[9]，ChristianRacca[10]，Andrea Bartolini[11]，Adrian Tate[12]，Berkay Yaman[13]

[1]哥廷根大学，德国哥廷根。[2]哥廷根科学数据处理中心（GWDG），德国哥廷根。[3]E4 Computer Engineering S.p.A.，意大利斯坎迪亚诺。[4]瑞典皇家理工学院，瑞典斯德哥尔摩。[5]HLRS，德国斯图加特。[6]华为技术有限公司杜塞尔多夫分公司，德国杜塞尔多夫。[7]Forschung Burgenland GmbH，奥地利艾森施塔特。[8]SYNYO GmbH，奥地利维也纳。[9]马尔马拉大学，土耳其伊斯坦布尔。[10]TOP-IX 联盟，意大利都灵。[11]博洛尼亚大学，意大利博洛尼亚。[12]Numerical Algorithms Group Limited，英国牛津。[13]BigTRI Bilisim A.S.，土耳其伊斯坦布尔。

贡献作者：julian.kunkel@gwdg.de；christian.boehme@gwdg.de；jonathan.decker@uni-goettingen.de；fabrizio.magugliani@e4company.com；pleiter@kth.se；koller@hlrs.de；karthee.sivalingam@huawei.com；Sabri.Pllana@forschungburgenland.at；alexander.nikolov@synyo.com；mujdat.soyturk@marmara.edu.tr；christian.racca@top-ix.org；a.bartolini@unibo.it；adrian.tate@nag.co.uk；berkay.yaman@bigtri.net；

摘要

DECICE是一个地平线欧洲计划项目，旨在开发一个由人工智能赋能的开放式、可移植管理框架，用于计算连续体中应用的自动化和自适应优化与部署，覆盖从边缘物联网传感器到大规模云／高性能计算基础设施。本文介绍DECICE框架及其架构。此外，我们还重点介绍用于评估该框架的应用案例：智能交通路口、磁共振成像和应急响应。

关键词： 人工智能赋能的计算连续体、云边协同编排、认知云、数字孪生、KubeEdge

1 引言

在过去十年中，云计算通过互联网提供能够根据用户需求进行弹性扩展的服务，彻底改变了计算领域。规模经济使得强大的计算资源和存储资源能够以更低的成本提供。此外，包括复制在内的容错机制能够显著提升部署在云中的服务的可用性。与此同时，智能家居、智慧城市和工业自动化等各种应用领域对计算服务提出了新的需求，要求将其部署在更靠近终端用户的位置。边缘计算通过将计算资源部署在网络边缘来补充云计算 [8]。它还支持连接使用 LoRaWAN 等替代性无线协议的物联网设备。

尽管云计算和边缘计算确实具有一些共同特征，但边缘计算也具备使其独具特色的特征和功能。将计算资源置于用户或数据源附近，可以降低延迟和数据传输量，甚至能够避免传输敏感数据。将计算资源部署在数据中心之外的趋势催生了计算连续体（或云连续体）这一术语 [14]。从网络边缘（以及更远的位置）一直延伸到云端和 HPC 数据中心的连续资源体系引起了越来越多的关注，在欧洲背景下也是如此 [5]。计算连续体的出现对于那些超出数据中心范围、必须利用能够跨联邦式基础设施访问分布式边缘、云和 HPC 资源的服务的工作流而言，尤其值得关注。

由于分布式组件之间存在相互依赖关系，且计算、存储和网络资源可能具有高度异构性，这些资源的调度成为一项巨大挑战 [3]。寻找最优调度决策的问题可能是NP完全。此外，故障、网络能力和系统拓扑的变化等动态因素，需要持续监控，并自动调整进程和数据的放置，因此，手动为进程创建良好的硬件静态分配将变得不可行。为应对这些挑战，我们认为有必要采用新的方法，其中包括

利用 AI 方法为拥有数千个节点的复杂系统及时制定资源分配决策。

本文介绍了 DECICE 项目 [9] ，该项目正在开发一个基于人工智能的、开放且可移植的云管理框架，用于在计算连续体中对应用程序进行自动化、自适应优化与部署。该框架依靠整体监控构建系统的数字孪生，以反映系统的所有组件及其特征。该数字孪生将向调度器提供信息，以便做出动态负载均衡和数据放置的调度决策。这将提高吞吐量并降低资源消耗。该框架的原型将集成到 Kubernetes 中，Kubernetes 是云环境中用于编排容器的最流行解决方案。Kubernetes 本身不内置对边缘计算的支持。为此，DECICE 依赖于 Kubernetes 的一个扩展，该扩展称为 KubeEdge [15] ，可将边缘节点透明地集成到主集群中。此外，为了支持在 Kubernetes 中执行类似 HPC 的批处理作业 [10] ，该原型将探索使用 Volcano 等技术 [4]。DECICE 项目正致力于实现以下目标:

- 基于开源和标准开发云管理框架，可在HPC、云和边缘连续体之间无缝连接并将应用部署到设备。
- 提供系统管理和 DevOps 工具，用于访问、控制和管理持续服务环境，该环境包括网络、计算、数据基础设施和服务的部署。
- 通过监控系统提供相关数据，为异构基础设施创建动态数字孪生，从而支持仿真和基于人工智能的预测。
- 借助人工智能模型自动执行应用调度决策，提升应用性能、可靠性、吞吐量、能效并降低成本。

本文其余部分结构如下。第2 节介绍背景信息。第3节概述DECICE框架。第4 节描述其架构。第5节介绍通过用例评估DECICE的策略。第6 节总结全文。

2 背景

对于云计算的兴起而言，提供虚拟化资源的能力至关重要。除虚拟机之外，容器已成为广泛使用的技术，而 Kubernetes 则是管理大规模容器执行的事实标准编排平台。Kubernetes 是一个开源容器编排平台，能够通过分配资源和虚拟化网络，动态处理跨数千个节点的容器化工作负载。Kubernetes 得以广泛使用的主要原因之一是其适应性，这使任何人都能够扩展、替换和移除这些系统的各个部分。

在云计算和容器编排解决方案出现之前，要扩展服务可用资源，需要采用专有解决方案来集成负载

均衡功能并添加额外的服务器。这些方法存在诸多缺陷，例如难以部署应用更新，也难以动态应对服务请求的突发激增。

容器提供的虚拟化简化了编排系统（如 Kubernetes）的运作：只要物理机器提供足够的资源，就可以选择在其上运行服务。这对于硬件同质且网络快速稳定的数据中心而言效果良好。但在包含不同类型硬件的环境中，情况就有所不同，而纳入边缘设备通常正是如此。

另一个挑战是，Kubernetes 是为云计算数据中心设计的，而非涉及边缘设备的分布式环境。KubeEdge [15], 应对了这一挑战，它扩展了 Kubernetes，使其能够使用边缘设备。KubeEdge 的主要特性包括为能力较弱的边缘设备提供轻量级 Kubernetes 安装，以及基于拉取的网络基础设施。借助这一网络基础设施，边缘设备可以作为常规节点出现在 Kubernetes 中。KubeEdge 的一项关键特性是实现了 OSI 网络协议层，从而能够将边缘设备和云服务器连接到同一个虚拟网络中。

随着边缘设备的加入，底层硬件资源集合变得异构，因此资源调度也显著复杂化。这里的调度是指基于当前可用资源利用率状况，决定将新容器部署在何处的算法。据我们所知，目前尚无解决方案能够应对由此产生的调度挑战，例如决定是否在云端或边缘节点上执行容器。为此，需要考虑可用边缘节点的具体硬件特性以及当前的网络连接情况。

容器调度是一个活跃的研究领域，目前正在探索各种技术。Ahmad 等人 [1] 将调度技术分为四类之一：(1) 数学建模技术；(2) 启发式技术；(3) 元启发式算法技术；(4) 机器学习技术。

数学技术为必须求解的优化问题提供数学描述，同时考虑一组约束条件。整数线性规划是一种广泛使用的技术。这类技术面临的一个典型挑战是：对于大规模系统而言，其复杂度会变得过高，求解时间也会过长。启发式技术通常具有良好的可扩展性且速度较快，但无法保证得到良好的调度决策。

元启发式算法和机器学习技术通常越来越多地用于与并行计算系统相关的优化问题（参见 [13] 中的最新综述）。元启发式算法技术基于搜索策略，用于针对给定的优化问题寻找较好的、但不一定是最优的解决方案。通常会使用受自然启发的算法 [6]。

机器学习技术使用在特定环境中收集的数据训练模型。之后，这些模型能够基于先前学习到的内容做出调度决策，例如利用模型预测所需容器的数量 [11]。

DECICE 计划采用机器学习技术进行异常检测。这类技术已成功应用于高性能计算系统（例如，

[2]）。此外，这类技术也已用于降低求解能耗（例如，[12]）。DECICE 将探索把此类方法扩展到计算连续体。

近期，数字孪生概念的受欢迎程度显著提升。该概念基于对一个或多个虚拟实体的实现，这些虚拟实体用于描述一个物理实体。数字孪生的一项关键特征是物理实体与虚拟实体的孪生，即实现物理到虚拟孪生和虚拟到物理孪生进程 [7]。据我们所知，数字孪生概念尚未应用于云边系统。

3 框架概述

DECICE 旨在开发一个基于人工智能的、开放且可移植的云管理框架，用于在联邦基础设施中自动执行应用程序的自适应优化与部署，其计算规模涵盖从超大型（例如 HPC 系统）到超小型（例如连接在边缘的物联网传感器）。

在如此悬殊的规模下运行，需要一个具备先进能力的智能管理平面，使其能够根据工作负载的需求（例如延迟、计算能力和功耗）主动调整系统内的工作负载。因此，我们设想构建一个人工智能模型，利用可用资源的数字孪生，根据资源的遥测数据做出实时调度决策。

DECICE框架能够动态平衡不同的工作负载，针对性能和能效优化系统资源（计算、存储和网络）的吞吐量与延迟，并快速适应不断变化的条件。该框架还为管理员和部署专家提供必要的工具和接口，使他们能够与所有基础设施组件交互并对其进行控制，以实现预期目标。

DECICE框架将通过开放API与编排系统集成，以实现可移植性、模块化和可扩展性。DECICE框架将通过具体用例进行协同设计和评估。

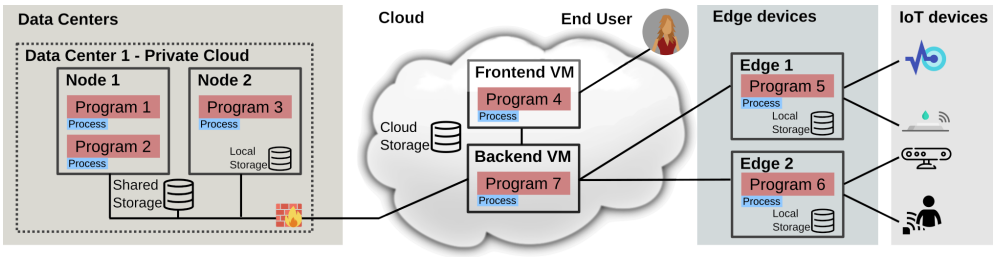


图1 计算连续体示例

图 1 所示的系统可视为一个执行非平凡应用程序的计算连续体实例。设有一个应用程序需要

在物联网设备上运行程序以收集数据，并通过两个边缘节点将数据传输至云节点。由于计算负载较重且存在隐私要求，数据中心可能会在私有云中提供计算资源和数据容量。随后，结果会传回云端，供最终用户使用。最后，云端提供前端界面，让最终用户能够访问应用程序和计算结果，例如通过网页访问。如何在整个计算连续体中分配数据是另一项尚待解决的挑战，计划在性能测试阶段加以处理。

4 DECICE 架构

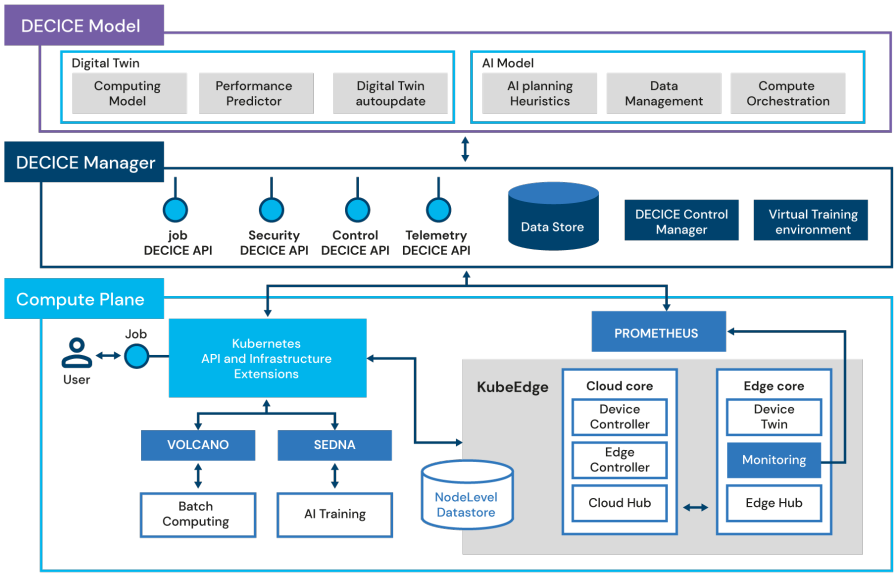


图 2 DECICE 总体架构

DECICE框架基于现有的开源技术构建，这些技术在社区中早已成熟并得到广泛应用。

Kubernetes、KubeEdge、Prometheus、SEDNA 和 Volcano 均属于 云原生计算基金会（*CNCF*）。该基金会是云原生生态系统中众多开源项目的中心，拥有 250 多家成员，其中包括许多公有云和企业云厂商。

将应用部署到云边环境的用户面临多重挑战，而且通常并不熟悉云管理框架或异构硬件资源。DECICE旨在帮助这类用户，尽可能减少他们为满足应用需求而高效利用多种硬件设备所需付出的努力。例如，DECICE框架的用户将能够使用标准容器构建混合边缘/云/HPC工作负载，并且可以指定数据局部性、网络延迟或其他约束；他们也知道，DECICE调度器会根据这些要求安排作业，并在需要时进行迁移。这些能力主要通过精心选择基础技术，并采用与DECICE交互的标准化接口来实现。更重要的是，通过对整个异构连续体进行恰当的映射和建模，将其纳入相应的DECICE模型中。

用户通过标准的 Kubernetes 接口与 DECICE 交互，该接口扩展了 DECICE 专用功能。这样，用户便可以使用现有的、支持完善且熟悉的接口。此外，这还使现有工作负载能够立即迁移到 DECICE，并最大限度地降低用户开始使用 DECICE 所需的“学习曲线”。

如图所示，计算平面、DECICEManager 和 DECICE模型将在以下章节中进行概述。2

4.1 计算平面

计算平面用于运行用户工作负载。它基于现有的开源技术构建，并与控制平面中的 DECICE 组件集成。计算平面的主要组件是 Kubernetes。它提供了在多个物理系统上运行容器化作业的基本构建模块。Kubernetes 还通过 KubeEdge 进行了扩展，以支持自主边缘设备的使用和管理。此外，Prometheus 用于通过 Kubernetes/KubeEdge 接口收集设备指标。最后，SEDNA 提供 AI 模型训练能力，而 Volcano 则支持运行 HPC作业。

4.2 DECICE管理器

DECICE管理器通过标准 API 将数字孪生和人工智能模型与编排系统（例如 Kubernetes）集成。这种集成可进行调整，使 DECICE框架能够跨编排框架移植，从而与任何商业或开源云解决方案集成。此外，DECICE管理器还包含一个数据存储库，用于存储基础设施和应用程序的元数据以及监控数据。这些数据用于更新数字孪生并训练人工智能模型。此外，管理器还提供合成训练与测试环境，用于模拟不同基础设施和应用程序的部署场景，以训练人工智能模型。

4.3 DECICE模型

动态数字孪生是在微观和宏观层面对实际或潜在系统的虚拟表示。它利用

仿真和实时监控对系统进行建模，并实现预测，再据此做出决策。创建异构云边基础设施及其已部署应用的数字孪生，是本项目的主要目标之一。DECICE数字孪生对计算、内存、存储和网络资源以及应用任务进行建模。仿真和实时监控用于持续更新模型。随后，人工智能模型用于预测新部署或发生故障时的系统行为。人工智能建模还用于根据系统当前状态持续预测应用性能，并提出优化或适应建议，以提高性能、能效、可靠性和吞吐量。随后由DECICE管理器执行这些建议。

4.4 虚拟训练环境

为了训练、改进和测试用于调度的人工智能模型，我们采用虚拟训练环境（VTE），该环境基于预先配置的场景模拟数字孪生（DT）。训练过程包括在训练场景中运行人工智能模型、更新场景，并根据一组指标评估人工智能调度器的性能。图 3 展示了这一过程：VTE 加载指定的训练场景，并据此配置 DT。在训练主循环中，VTE 通过推进场景来推进仿真，进而向 DT 提交更多指标和作业更新。随后，调度器访问 DT 并执行调度决策。

由于 VTE 负责在仿真中推进场景，而人工智能调度器本身并不具备历史状态的概念，因此可以进行快于实时的训练。这使我们能够快速并行训练多个人工智能调度器实例，并保留表现最佳的实例以供进一步训练。

图 3 中 VTE 使用的训练循环与将在真实系统上部署的调度循环类似。真实系统由云管理器（CM）控制，而非 VTE；DT 的更新则从指标系统（MS）和 CM 获取。此外，CM 请求的调度决策随后会转发到底层 Kubernetes（K8s）平台并予以执行。

值得注意的是，人工智能调度器始终查看整个 DT，其中也包含完整的作业队列以及所有运行中作业的状态。人工智能调度器没有义务立即调度新到达的作业，其目标是在遵循用户规范的同时，优化底层集群对所有活动作业和排队作业的利用率。这还包括定期重新评估系统当前状态，并在出现更佳放置方案时重新调度运行中的作业。

总体而言，人工智能调度器能够通过数字孪生查看并优化整个集群。常见的基于启发式方法的调度器会逐一评估队列中的每个作业，但也可以扩展为一次查看所有作业。然而，据我们所知，这会大幅增加所用启发式方法的复杂度，却无法从更深入地理解系统各组件之间的相互作用中获益，而基于机器学习的调度器则能够做到这一点。我们还将探索各种深度学习策略。

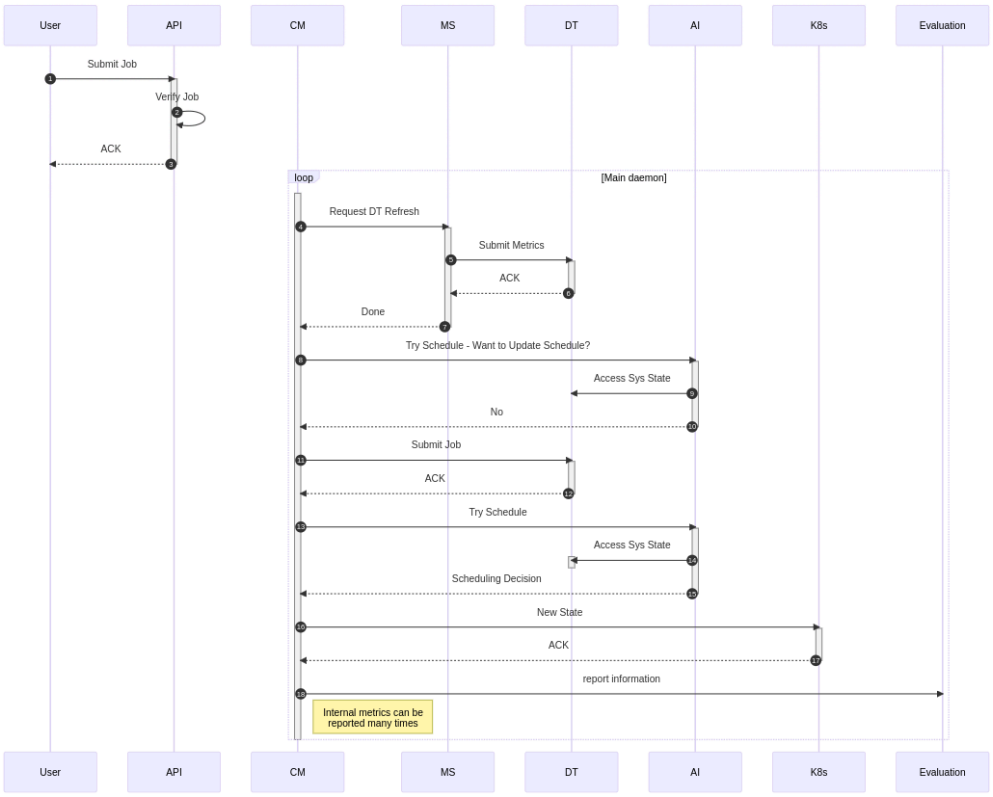


图 3 时序图：虚拟化训练；VTE：虚拟训练环境；DT：数字孪生

4.5 CI/CD 测试策略

我们的 CI/CD 系统旨在验证所有组件单独运行及组合运行时是否均能正常工作。在 DECICE 项目中开发的组件，例如云管理器、数字孪生和人工智能调度器，均通过单个组件测试和模拟测试验证其逻辑。

由于 DECICE 系统与底层编排平台集成，而该平台也会独立于 DECICE 随其生态系统不断发展，因此有必要定期执行集成测试。目前，DECICE 系统的原型仅面向 Kubernetes 这一编排平台。

根据定义，DECICE 系统应可供管理包含云、边缘和 HPC 资源的异构计算系统的管理员使用。管理员可能还有其他要求，例如在其系统上使用特定版本的 Kubernetes，该版本可能与开发 DECICE 时使用的版本不同。因此，DECICE 的集成测试涵盖了一系列常用版本，以确保兼容性。

各个组件通过单元测试和模拟测试进行测试。不过，集成测试需要先启动一个临时 Kubernetes 集群，并将组件部署到其中，然后才能对它们进行测试。这样便可在多轮测试中使用不同的配置、版本和其他元素（例如边缘节点和高性能计算节点）。

在实践中，我们考虑使用 GitLab 运行器处理 CI/CD 流水线。它们可以与 Terraform 和 OpenStack 集成，以自动部署虚拟机、在其上安装 Kubernetes，并将 DECICE 系统部署其中。这样的系统使我们能够针对多个版本的 Kubernetes、KubeEdge 和 Volcano，对软件进行完整的端到端测试。

5 评估策略

本节介绍我们对 DECICE框架 的评估方法。DECICE 项目采用协同设计流程，以确保 DECICE 解决方案的功能和性能能够满足现实挑战的需求。图 4 展示了用于演示本项目所开发技术能力的应用场景，

- 智能交叉路口
- 磁共振成像 (MRI)
- 支持应急响应的现场智能

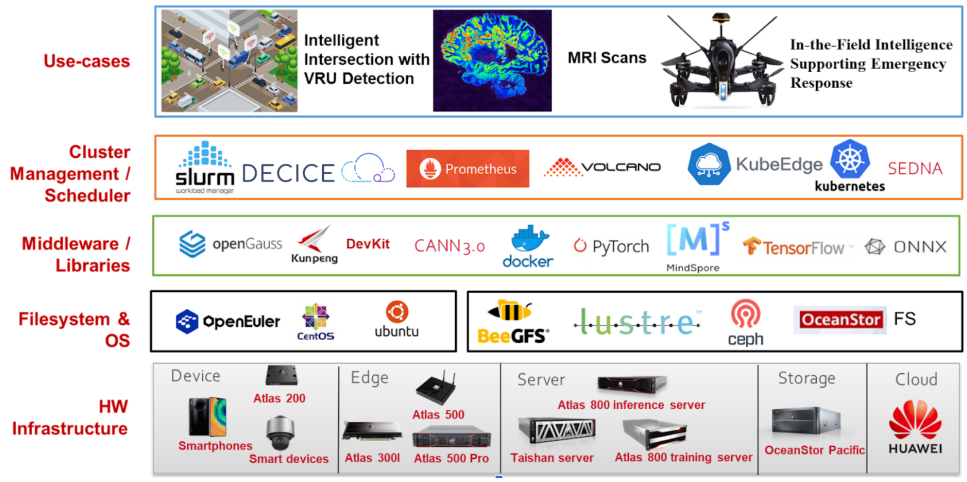


图 4 DECICE 评估应用场景：智能交叉路口、MRI 和应急响应

5.1 智能交叉路口

此用例聚焦于网联与自动驾驶及协作式智能交通系统（C-ITS）中的实时处理方面。网联自动驾驶汽车（CAVs）利用车载传感器进行目标检测和态势感知。由于不同数量的道路使用者（汽车、卡车、公交车、行人、自行车、摩托车、轮椅使用者等）朝不同方向行驶，交叉路口会带来额外的难题、挑战和安全问题。检测道路使用者（包括弱势道路使用者（VRU））对于确保安全和交通效率至关重要。被其他物体或拐角遮挡的道路使用者可能无法被车辆上的车载传感器检测到。这会对道路安全、驾驶安全和交通效率构成风险。除了共享车辆状态信息外，还旨在实时向道路使用者（车辆、弱势道路使用者等）共享车辆及交叉路口传感器（如摄像头）获取的信息。C2C-CC 将此方法纳入第二阶段和 Day3+应用。

部署在交叉路口的摄像头持续监控感兴趣区域（RoI），并对捕获的图像/视频进行分析，以检测目标对象（车辆、弱势道路使用者（VRU）等），随后必须将检测结果共享给道路使用者（车辆、弱势道路使用者）。图像处理和通信必须实时进行（保持低延迟），以保障安全并避免潜在事故。

由此产生的问题是，应在哪个层级处理图像。这可以在云端或边缘侧完成，具体取决于相关需求及实施成本。准确率、延迟和效率是衡量技术性能的关键指标。摄像头与中央云之间的数据通信会引入额外的（通信）延迟，这是不可接受的。因此，总体目标是在边缘侧执行一些需要快速处理、低延迟和快速响应的应用，而学习模型则可以在边缘侧和云端运行。另一方面，资源利用率、成本和交通安全则是与业务相关的性能指标。DECICE 提供了一种高效、快速处理、低延迟且节能的边缘与云计算连续体，同时还考虑了用户（车辆、弱势道路使用者）的时空变化以及环境条件（光线、照明、天气状况等）所导致的边缘节点负载。

5.2 磁共振成像（MRI）

MRI扫描因其能够以非侵入方式获取体内器官的影像，已成为医学检查中的常用工具。通常，MRI扫描在边缘节点上进行；扫描仅覆盖患者的单个身体部位，并利用磁场生成该部位的高分辨率三维图像。技术进步带来了更高的分辨率，进而产生更大的数据量。与其他医学数据一样，MRI扫描通常包含患者的敏感个人信息。因此，必须特别关注数据的存储、传输和访问。我们提出将MRI扫描分析作为DECICE框架的一个用例，以验证对数据存储和访问权限的限制。此外，对MRI扫描进行快速且准确的分析十分重要，因此快速的计算时间也被列入需求之中。

MRI扫描的分析在计算和存储方面都面临困难。每次MRI扫描由多个数据切片组成，因此形成了一个复杂的分析对象。用于分析的复杂模型可能会超出边缘设备的计算能力。因此，可能有必要将分析作业迁移到云或HPC系统中。从存储角度来看，管理数据访问权限、数据存储位置以及防止数据丢失是关键环节。此外，网络传输可能成为限制因素，从而形成复杂的决策环境。DECICE调度器将把数据和任务映射到合适的位置，并适应系统拓扑结构的变化，而不是让用户手动寻找解决方案。

5.3 支持应急响应的现场智能

在应急响应中，任务区域内的计算基础设施和连接可能会受到严重干扰，变得不可靠。为了持续监测受影响区域，并在灾害发生后有效响应，可以借助无人机：整合无人机上多个传感器的数据，以及卫星图像等其他数据源，以确定任务区域或扩大关注范围。四旋翼无人机很有应用前景，但任务续航时间有限。也可以在机载设备上处理图像，但必须适应不同条件，因此计算方式需要具备灵活性。可使用充电站延长无人机的自主运行时间，同时支持上传任务计划、收集数据和通信。任务可根据天气情况进行调整。操作人员与无人机巡查之间需要协调，以覆盖操作人员无法覆盖的区域。处理过程必须经过优化；还可以整合卫星数据等外部数据源，以完善任务。

我们的目标是开发一个开放的数字平台，为应急响应人员提供支持，使其能够利用无人机和卫星数据。该平台旨在通过提供计算支持，将机器学习算法嵌入现场作业，从而为无人机自主飞行和任务执行提供支持。为此，平台将以高效、自适应的方式在现场计算人工智能算法，以容错方式协调云、边缘与无人机之间的处理和通信，使操作人员能够控制各计算实体之间的数据流，并提供系统平台所需的自适应能力，以应对各种紧急情况，例如待探索区域的大小、需要执行的识别任务、通信链路带宽、能源可用性和天气状况。

6 结论

过去十年，云计算行业大幅增长，随之也涌现出新的应用领域。现代云应用也变得更加复杂，因为它们通常运行在分布式计算机系统上，并被拆分为必须保持高可用性的组件。

在 DECICE 项目中，我们的目标是开发一个开放且可移植的云边管理框架，通过将作业映射到异构系统环境中最合适的资源，实现应用程序的自动化和自适应优化，并

计算连续体。通过利用整体监控，我们构建了系统的数字孪生，以反映原始系统的状态。人工智能调度器负责决定作业和数据的放置，并执行作业重新调度，以动态适应系统变化。我们开发了一个虚拟训练环境，用于生成测试数据，以训练机器学习模型并探索假设场景。该可移植框架集成到 Kubernetes 生态系统中，并通过在真实世界的异构系统上采用相关用例进行协同设计和评估。

致谢

本研究是在 DECICE 项目背景下开展的，该项目由欧盟资助（资助协议编号：101092582）。

参考文献

[1] Ahmad, I., AlFailakawi, M. G., AlMutawa, A., and Alsalman, L. 容器调度技术：综述与评估。《沙特国王大学学报——计算机与信息科学》（2021年3月）。[2] Borghesi, A., Molan, M., Milano, M., and Bartolini, A. 高性能计算系统中的异常检测与预判。《IEEE 并行与分布式系统汇刊》33, 4（2022年），739–750。[3] Carrión, C. Kubernetes 调度：分类法、现存问题与挑战。《ACM 计算调查》55, 7（2022年12月）。[4] CNCF。Volcano：面向计算密集型工作负载的云原生批处理调度系统。https://volcano.sh/en/。访问日期：2023年11月3日。[5] ETP4HPC。跨连续体倡议愿景。https://www.etp4hpc.eu/tci-vision.html。访问日期：2023年10月2日。[6] Ferrandi, F., Lanzi, P. L., Pilato, C., Sciuto, D., and Tumeo, A. 用于异构嵌入式系统中任务与通信映射和调度的蚁群启发式算法。《IEEE 集成电路与系统计算机辅助设计汇刊》29, 6（2010年），911–924。[7] Jones, D., Snider, C., Nassehi, A., Yon, J., and Hicks, B. 数字孪生特征研究：系统性文献综述。《CIRP 制造科学与技术杂志》29（2020年），36–52。[8] Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., and Ahmed, A. 边缘计算：综述。《未来一代计算机系统》97（2019年），219–235。[9] Kunkel, J., Boehme, C., Magugliani, F., Pleiter, D., Koller, B.,

Sivalingam, K., Pllana, S., Nikolov, A., Soytuerk, M., Racca, C., Bartolini, A., Tate, A., and Yaman, B. 设备-边缘-云智能协作框架（DECICE）。https://cordis.europa.eu/project/id/101092582，2022年。访问日期：2023年10月2日。[10] Liu, P., and Guitart, J. Kubernetes集群中容器化高性能计算工作负载的细粒度调度。arXiv预印本 arXiv:2211.11487（2022年）。[11] Lv, L., Zhang, Y., Li, Y., Xu, K., Wang, D., Wang, W., Li, M., Cao,X., and Liang, Q. 大型互联网数据中心中通信感知的容器放置与重新分配。IEEE通信领域精选期刊 37，3（2019年），540–555。[12] Memeti, S., and Pllana, S. 结合人工智能规划启发式算法和机器学习的异构系统优化：一种性能与能耗感知方法。《计算》103，12（2021年12月），2943–2966。[13] Memeti, S., Pllana, S., Binotto, A., Kołodziej, J., and Brandic, I.使用元启发式算法和机器学习进行并行计算系统的软件优化：系统性文献综述。《计算》（2018年4月）。[14] Moreschini, S., Pecorelli, F., Li, X., Naz, S., Hästbacka, D., and Taibi,D. 云连续体：定义。IEEE Access 10（2022年），131876–131886。[15] Xiong, Y., Sun, Y., Xing, L., and Huang, Y. 使用KubeEdge将云扩展至边缘。载于2018年IEEE/ACM边缘计算研讨会（SEC）（2018年10月），第373–377页。