

云边端协同平台在智能制造中的应用与优化

苗晓培¹ 罗志伟¹
MIAO Xiaopei LUO Zhiwei

摘要

探讨了云边端协同平台在智能制造中的应用与优化,提出了一种基于二阶差分的启发式算法,用于解决云边协同计算架构下IO密集型虚拟机的资源分配问题。同时,针对智能制造任务的复杂性,设计了CECTS(云边缘协作任务调度)算法,实现了制造任务在异构云边环境下的高效执行。通过仿真实验,验证了所提算法在任务完成时间、超时率和资源利用率等方面的优越性。研究成果为推动智能制造的创新发展提供了重要的技术支持。

关键词

云边端; 协同平台; 智能制造

doi: 10.3969/j.issn.1672-9528.2024.10.048

0 引言

云边端协同平台可以支持制造过程的实时监控和快速决策,迅速响应生产线上的异常情况,基于全局数据提供优化决策建议。然而,在异构的云边端环境中,如何高效分配计算资源、合理调度复杂的制造任务,成为亟待解决的技术难题。本研究旨在探索这些问题的解决方案,为制造业的数字化转型和智能化升级提供技术支持。

1 云边端协同平台在智能制造体系中的应用路径

随着工业4.0和智能制造的快速发展,云边端协同计算架构正在成为支撑智能制造系统的关键技术框架。云边端协同平台可以有效支持智能制造中的大规模数据采集和处理,保证数据处理的实时性,提供强大的分析能力^[1]。而且通过边缘计算节点的就近处理能力,云边端协同平台可以支持制造过程的实时监控和快速决策,迅速响应生产线上的异常情况,基于全局数据提供优化决策建议。此类平台还能够为智能制造提供灵活的资源调度能力,系统可以制定全局生产计划,并利用边缘节点根据实时状况进行局部调整,从而实现生产的柔性化和个性化^[2]。

这些应用体系都需要完善资源分配和任务调度,才能真正落实到智能制造领域。因此,需要积极探索如何在异构的云边端环境中高效分配计算资源、合理调度复杂的制造任务。通过探索这些应用路径,云边端协同平台有望成为推动智能

制造创新发展的重要技术支持,为制造业的数字化转型和智能化升级提供强有力的辅助功能。

2 云边协同计算架构下智能化制造资源分配方案设计

2.1 资源分配模型

云边协同计算架构下,智能制造工业经常会面临IO密集型虚拟机的资源分配问题,研究基于常用架构,提出一种资源分配模型。考虑到边缘端服务器需要处理大量的终端设备数据,传输的数据类型多样,且不同类型数据对IO虚拟机资源需求不同,在满足终端设备服务质量的前提下,应着重提高边缘服务器的资源利用率,尽可能减少出现资源竞争和等待的情况,因此可给出如公式(1)所示的资源分配模型:

$$\max \sum_{j=1}^m \frac{U_j(t)}{S_j}, \quad \text{s.t.} \quad \sum_{j=1}^m S_j \leq C \quad (1)$$

式中: m 是边缘服务器上IO虚拟机的数量, C 是边缘服务器可分配给IO虚拟机的总容量。目标函数 $\sum_{j=1}^m \frac{U_j(t)}{S_j}$ 表示IO虚拟机整体的资源利用率,约束条件限制所有虚拟机预分配空间之和不超过边缘服务器的总容量^[3]。

为求解该模型,可以考虑启发式的资源分配算法。根据数据权重 w_i 和历史数据传输量,预测一段时间内各类数据的到达量,然后结合虚拟机的动态占用量 $U_j(t)$,触发资源调整,将空闲资源优先分配给数据量大、权重高的虚拟机。同时,需要权衡资源分配的公平性,避免出现个别虚拟机长期得不到服务的情况^[4]。

2.2 基于二阶差分的启发式算法

为了高效求解上文提出的IO虚拟机资源分配模型,研究设计了一种基于二阶差分的启发式算法。该算法利用虚拟

1. 广州城建职业学院 广东广州 510920

[基金项目] 广东省普通高校特色创新类项目“面向智能制造领域的基于云、边、端协同应用机制研究”(2023KTSCX412)

机实际占用空间在时间维度上的变化特征,通过二阶差分方程预测虚拟机的资源需求趋势,从而实现动态的资源分配^[5]。

考虑在离散时间下虚拟机占用空间的变化过程,可引入二阶差分方程,其公式为:

$$\begin{aligned}\Delta^2 U_j(t) &= U_j(t+2) - 2U_j(t+1) + U_j(t) \\ &= \lambda (\Delta U_j(t+1) - \Delta U_j(t))\end{aligned}\quad (2)$$

式中: $\Delta U_j(t) = U_j(t+1) - U_j(t)$, 以及方程右端 $\Delta U_j(t+1) - \Delta U_j(t)$ 表示占用空间变化趋势的加速度, 参数 λ 反映加速度的放大或缩小程度。将方程变形, 得到含参数 λ 的预测模型, 其公式为:

$$U_j(t+2) = (2 + \lambda)U_j(t+1) - (1 + \lambda)U_j(t) \quad (3)$$

通过最小二乘法拟合历史数据, 可以估计出参数 λ , 代入上式即可预测未来的 $U_j(t+2)$ 值。基于占用空间预测值 $U_j(t+2)$, 可以判断虚拟机的资源需求趋势。若 $U_j(t+2) > U_j(t+1)$, 则意味着虚拟机 j 的资源需求呈增长态势, 应当给予更多资源; 反之, 若 $U_j(t+2) < U_j(t+1)$, 则说明虚拟机资源需求下降, 可以适当回收其空闲资源。令 $\Delta S_j(t)$ 表示 t 时刻分配给虚拟机 j 的资源调整量, 资源分配决策公式为:

$$\Delta S_j(t) = \begin{cases} \alpha(U_j(t+2) - U_j(t+1)), & \text{if } U_j(t+2) > U_j(t+1) \\ \beta(U_j(t+2) - U_j(t+1)), & \text{if } U_j(t+2) < U_j(t+1) \end{cases} \quad (4)$$

式中: $0 < \beta < 1 < \alpha$, 表示资源调整的力度因子。 $\alpha > 1$ 鼓励资源向需求增长的虚拟机倾斜, 而 $0 < \beta < 1$ 避免过度回收空闲资源, 以应对需求的突发波动。将资源调整量 $\Delta S_j(t)$ 叠加在原空间分配 S_j 上, 可得虚拟机 j 在 t 时刻新的空间分配, 其公式为:

$$S_j^{\text{new}}(t) = S_j + \Delta S_j(t) \quad (5)$$

为体现资源调整的公平性, 令 $\delta_j(t) = \frac{U_j(t)}{S_j}$ 表示虚拟机 j 在 t 时刻的资源利用率。在资源调整时, 优先考虑满足利用率高的虚拟机的增长需求, 避免持续低利用率的虚拟机占据过多资源。引入公平因子 $\gamma_j(t) = e^{\delta_j(t) - \bar{\delta}(t)}$, 其中 $\bar{\delta}(t)$ 是所有虚拟机 t 时刻利用率的平均值。将公平因子代入资源分配决策中, 得到最终的资源调整量, 其公式为:

$$\Delta S_j^{\text{final}}(t) = \begin{cases} \alpha \gamma_j(t) (U_j(t+2) - U_j(t+1)), & \text{if } U_j(t+2) > U_j(t+1) \\ \beta \gamma_j(t) (U_j(t+2) - U_j(t+1)), & \text{if } U_j(t+2) < U_j(t+1) \end{cases} \quad (6)$$

该算法基于二阶差分方程预测虚拟机的资源需求变化趋势, 兼顾资源分配的动态性与公平性, 可以提升云边协同场景下的资源利用效率。在实际系统中, 需要对各项参数进行调优, 以平衡算法的预测精度和计算开销。

3 云边协同计算架构下智能化制造任务调度方案设计

智能制造场景下, 生产任务呈现出数量多、类型复杂、

关联性强等特点。单独的云端或边缘节点难以高效处理全部任务, 因此需要设计协同的任务调度方案, 将任务合理分配到云、边、端多层计算资源上^[6]。研究在智能制造任务 DAG 模型的基础上, 提出一种 CECTS (cloud-edge collaborative task scheduling, 云边缘协作任务调度) 算法, 通过任务拆分、节点选择、任务排序步骤实现制造任务在异构云边环境下的高效执行。

智能制造过程通常包含多个关联生产任务, 每个任务可表示为一个有向无环图 (DAG)。DAG 节点代表子任务, 边代表任务间的依赖关系和数据传输。多个 DAG 可能存在公共子任务或数据交互。因此, 需要应用多 DAG 合并与分割策略, 优化任务粒度, 减少跨节点的数据传输。对于不同 DAG 的公共子任务, 可以将其合并为一个任务节点, 避免重复计算。例如, DAG1 和 DAG2 都包含子任务 T3, 则将两个 T3 节点合并为一个, 同时更新关联边。合并后的公共任务可以被多个父任务共享, 降低总体计算量。如果两个 DAG 之间存在数据依赖, 即一个 DAG 的输出是另一个 DAG 的输入, 可以通过拆分数据依赖边来解耦 DAG。将依赖边上的数据传输任务独立为一个新的子任务节点, 连接原有的两个任务节点。拆分后的 DAG 之间通过数据传输任务连接, 减少了直接依赖。

在任务调度过程中, 既要避免任务粒度过大造成负载不均衡, 也要避免任务粒度过小导致频繁的数据传输。因此, 需要对合并后的 DAG 进一步优化任务粒度。可以设置任务粒度阈值, 对小于阈值的任务节点进行合并, 直到合并后的任务粒度达到阈值, 粒度阈值的选取则需要综合考虑计算资源的异构性和任务的并行度。通过以上策略, 可以将智能制造任务的多个 DAG 转化为粒度合适、依赖关系简化的 DAG, 为后续的任务调度提供更好的输入, 如图 1、图 2 所示。

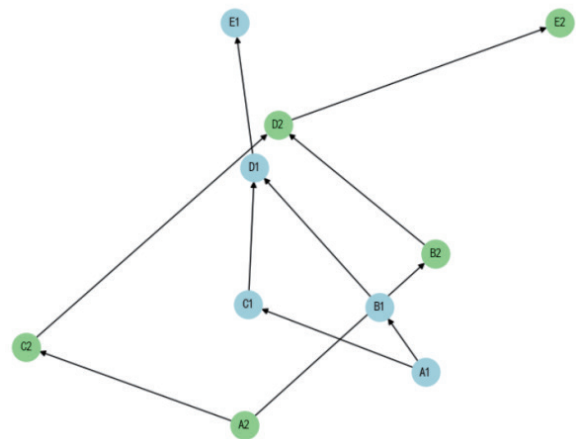


图 1 原始多 DAG

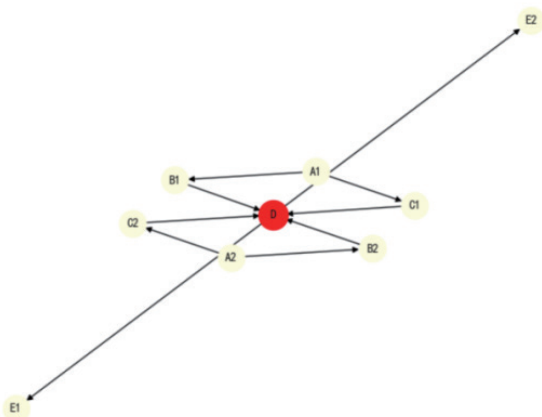


图2 优化后 DAG

在获得优化后的制造任务 DAG 后，可以应用 CECTS 调度算法，将任务节点映射到云边异构计算资源上，并确定任务的执行顺序。同一计算资源上，应按照任务优先级对任务执行顺序进行优化，尽可能减少优先级高的任务等待时间。并且考虑任务间的数据依赖，保证依赖任务的完成先于被依赖任务。

通过以上策略，CECTS 算法可以根据制造任务的特点和云边资源的异构性，生成高效的任务调度方案。与传统的启发式调度算法相比，CECTS 更全面地考虑了任务优先级、资源选择和执行顺序优化，能够有效减少任务的平均完成时间，提高任务调度 QoS 保障能力^[7]。

4 仿真实验与性能评估

为验证算法的有效性，研究从某工业企业收集连续 1 个月的生产任务数据，包括任务工序、资源需求、执行时间信息^[8]。每个任务对应一个 DAG，平均包含 20 个子任务，最大子任务数为 50。子任务的计算量服从正态分布，数据依赖量服从泊松分布。为体现任务的并发性，实验每隔 10 min 随机提交 5 ~ 10 个任务^[9]。选取了 3 种典型的任务调度算法作为对比，即 HEFT 静态列表调度算法、DBUS 动态调度算法以及 EAJS 基于进化算法的调度优化方法^[10]。

评估结果发现，从任务完成时间来看，CECTS 算法相比其他算法有明显优势，平均完成时间最短。HEFT 和 EAJS 的完成时间波动较大，说明静态调度难以应对动态任务的即时性需求。任务超时率方面，CECTS 同样表现最优，平均超时率控制在 5% 以内。相比之下，DBUS 和 EAJS 的超时率较高，HEFT 的超时率最高。资源利用率方面，CECTS 在保证任务 QoS 的同时，也兼顾了资源利用的均衡性，平均利用率接近 80%。DBUS 利用率最高，但可能以牺牲任务 QoS 为代价。HEFT 和 EAJS 的利用率则相对较低。从算法本身的效率分析，CECTS 的运行时间相对 DBUS 和 HEFT 略长，但在可接受

的范围内，平均仅为任务完成时间的 1%。相比之下，EAJS 由于采用了进化算法，运行开销较高，平均是 CECTS 的 5 倍以上。

5 总结

本研究设计了一种基于二阶差分的启发式算法，用于解决云边协同计算架构下 IO 密集型虚拟机的资源分配问题。该算法通过预测虚拟机的资源需求趋势，实现了动态和公平的资源分配。通过仿真实验，验证了所提算法的有效性。结果表明，与传统调度算法相比，CECTS 算法在任务完成时间、超时率和资源利用率等方面都表现出明显优势。未来需考虑更复杂的制造场景，并探索将人工智能技术融入资源分配和任务调度过程中。

参考文献：

[1] 倪宏秋,谈海生,郑子木.云边端协同在线智能计算[J].自动化博览,2023,40(10):42-45.

[2] 谷思远,罗雪山,蒋凯,等.基于云边端协同的轻量级作战资源弹性调度算法设计与实现[J].火力与指挥控制,2023,48(10):122-132.

[3] 刘洋,于海东,刘文彬,等.基于“云边端”协同的分布式光伏柔性调节体系架构[J].山东电力技术,2023,50(7):23-29.

[4] 周旭,李琢.面向算力网络的云边端协同调度技术[J].中兴通讯技术,2023,29(4):32-37.

[5] 柳义筠,陈善平.软件定义车联网云边端协同一体化架构的研究[J].计算机时代,2023(4):23-28.

[6] 尚玉叶,袁家斌.深空环境中基于云边端协同的任务卸载方法[J].计算机科学,2023,50(2):80-88.

[7] 姜成中,王小奇.打造云边端协同一体化边缘智能平台助力智慧海关建设[J].中国新通信,2022,24(17):70-72.

[8] 徐帅帅,苏敏杰,任迅,等.云边端协同下的任务调度与资源分配方法[J].电信工程技术与标准化,2024,37(6):50-56.

[9] 谢满德,黄竹芳,孙浩.云边端协同下多用户细粒度任务卸载调度策略[J].电信科学,2024,40(4):107-121.

[10] 宋今,张河翔.云边端融合的算网架构在智慧园区中的应用研究[J].信息通信技术与政策,2024,50(2):55-62.

【作者简介】

苗晓塔（1970—），男，湖北荆州人，硕士，副教授，研究方向：智能控制技术、云计算技术。

罗志伟（1992—），男，广东广州人，硕士，助教，研究方向：电机控制与驱动、机器人技术。

（收稿日期：2024-07-03）