

智能配置与负载感知调度的融合主机虚拟资源优化

齐玉玲¹, 黄涛¹, 刘国菲¹, 张军贤¹, 鲍春晓², 吴江鹏², 黄宜华³

1. 中车南京浦镇车辆有限公司, 江苏 南京 210003;
2. 江苏鸿程大数据技术与应用研究院有限公司, 江苏 南京 210003;
3. 计算机软件新技术国家重点实验室(南京大学), 江苏 南京 210023

摘要

针对边缘计算平台中因异构应用共存、动态负载频发引发的虚拟化资源配置不准确和调度效率低的问题, 以列车融合主机为典型应用场景, 提出一种基于智能化配置与负载感知调度的融合主机虚拟化资源联合优化方法。首先, 通过量化分析融合主机典型应用的资源需求特征, 构建了一种基于随机森林建模和二分查找法的资源配置预测模型, 实现对虚拟化资源的精准前瞻性分配。其次, 针对动态负载变化, 设计了一种改进的遗传算法, 该算法将虚拟化应用与物理 CPU 核心进行映射, 并结合资源利用率与应用性能的多目标适应度函数, 动态调整调度策略。实验结果表明, 与传统优势资源公平 (dominant resource fairness, DRF) 算法相比, 所提出的资源配置预测模型能提供优于人工初始化的配置参数, 同时改进的遗传算法将 CPU 平均利用率从 13.5% 提升至 22.07%, 相对提升幅度达 63.5%, 目标函数值从 0.035 提升至 0.204, 提升约 4.83 倍, 服务器总资源占用降低 44%, 有效节约了硬件成本与能耗开销。研究为边缘计算平台在高动态场景下的资源优化提供了通用方法, 并以列车融合主机为例验证了其可行性, 对智能边缘系统的构建具有普适参考价值。

关键词

边缘计算平台; 虚拟化技术; 资源配置; 动态调度; 遗传算法; 随机森林

中图分类号: TP302.7

文献标志码: A

doi:10.11959/j.issn.2096-0271.2026014

Intelligent configuration and load-aware scheduling for optimizing integrated host virtual resource

Qi Yuling¹, Huang Tao¹, Liu Guofei¹, Zhang Junxian¹, Bao Chunxiao², Wu Jiangpeng², Huang Yihua³

1. CRRC Nanjing Puzhen Co., Ltd., Nanjing 210003, China
2. Jiangsu Hongcheng Big Data Technology and Application Research Institute Co., Ltd., Nanjing 210003, China
3. State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing 210023, China

Abstract

Addressing the issues of inaccurate virtualized resource allocation and low scheduling efficiency caused by the coexistence of heterogeneous applications and frequent dynamic loads in edge computing platforms, a joint

optimization method for virtualized resources of a converged host based on intelligent configuration and load-aware scheduling was proposed, using a train-related converged host as a typical application scenario. Firstly, by quantitatively analyzing the resource demand characteristics of typical applications of the converged host, a resource allocation prediction model based on random forest modeling and binary search method was constructed to achieve precise and forward-looking allocation of virtualized resources. Secondly, in response to dynamic load changes, an improved genetic algorithm was designed. This algorithm mapped virtualized applications to physical CPU cores and dynamically adjusted the scheduling strategy by combining a multi-objective fitness function that considers resource utilization and application performance. Experimental results showed that, compared to the traditional dominant resource fairness (DRF) algorithm, the resource allocation prediction model proposed provided configuration parameters superior to those initialized manually. Meanwhile, the improved genetic algorithm simultaneously increased the average CPU utilization from 13.5% to 22.07%, representing a relative increase of 63.5%. The objective function value increased from 0.035 to 0.204, a approximately 4.83-fold increase, reducing the total server resource consumption by 44%, and effectively saved hardware costs and energy consumption. The study provides a general method for resource optimization in edge computing platforms under highly dynamic scenarios, and verifies its feasibility using the train converged host as an example. It has universal reference value for the construction of intelligent edge systems.

Key words

edge computing platform, virtualization technology, resource configuration, dynamic scheduling, genetic algorithm, random forest

0 引言

边缘计算平台凭借其低时延、高可靠的特点，正逐步深入各垂直行业。同时，现代列车系统正向着高度集成化与智能化的方向演进^[1]。云计算、边缘计算、智能终端深度融合的云边端协同环境已成为支撑数智化发展的重要信息基础设施^[2]。为降低边缘计算平台的硬件成本、体积与功耗，提升部署灵活性，本文以列车融合主机为应用场景展开具体研究，通过在单一物理硬件平台上运行多个虚拟机（virtual machine, VM）来承载不同的子系统功能^[3-6]，并分析其资源应用特性。

然而，虚拟化技术在带来诸多优势的同时，也引入了新的技术挑战。当前，边缘平台资源管理主要面临两大瓶颈。一是资源配置静态化。管理员通常根据经验或应用峰值需求进行手动配置，这种方式难

以精确匹配应用的实际需求，常导致部分资源闲置甚至是资源分配失衡现象，既浪费了宝贵的硬件资源，也可能因资源不足形成性能瓶颈。二是调度策略通用化。传统的虚拟机调度算法（如完全公平调度器（completely fair scheduler, CFS）算法^[7]等）对车载应用的异构特性感知不足，在处理计算密集型、输入输出（input/output, I/O）密集型和实时性要求高的混合负载时，容易因资源竞争引发关键任务响应时延超标，威胁行车安全^[8]。

针对上述问题，国内外学者开展了一系列研究。当前，基于优势资源公平（dominant resource fairness, DRF）算法^[9-14]进行多维资源调度时，应用初始资源配置采用基于静态模板的资源划分策略，保障了应用的隔离性，但在面对动态负载时适应性不足，同时忽略了不同应用间的优先级差异。这些研究为虚拟化资源优化提供了思路，但普遍存在两个不足：

一是缺乏对列车特定应用负载特征的深度挖掘与量化建模；二是资源配置与动态调度环节相对独立，未能形成协同优化的闭环。

为解决上述不足，本文提出一种基于智能化配置与负载感知调度的融合主机虚拟化资源联合优化方法。该方法的核心贡献如下。

(1) 构建了一个基于随机森林的应用异构特征感知的资源配置模型，能够根据应用的静态属性与动态负载特征，预测其资源需求。

(2) 设计了一种负载动态调整的遗传调度策略，通过多目标优化函数引导调度方案的演进，实现资源利用率 and 应用性能的平衡。

最终，该方法通过配置与调度的协同作用，实现融合主机虚拟化资源的全局优化。

1 智能资源调度框架设计与系统架构

为实现对融合主机虚拟化资源的合理配置与调度优化，本文提出的联合优化方法系统架构如图1所示。该系统主要由智能化虚拟子系统资源配置模型和负载感知优化虚拟子系统调度策略两大核心模块组成，二者通过数据与决策的闭环进行联动，其中将融合主机上的虚拟机称为子系统。

联合优化方法的具体流程如下。

(1) 数据采集与特征提取

首先，通过部署在宿主机上的监控代理，实时采集各个虚拟化子系统的性能数据，包括CPU利用率、内存使用量、网络带宽和I/O吞吐量等。同时，提取子系统的静态属性，如任务类型（计算密集型、I/O密集型）、业务优先级等。

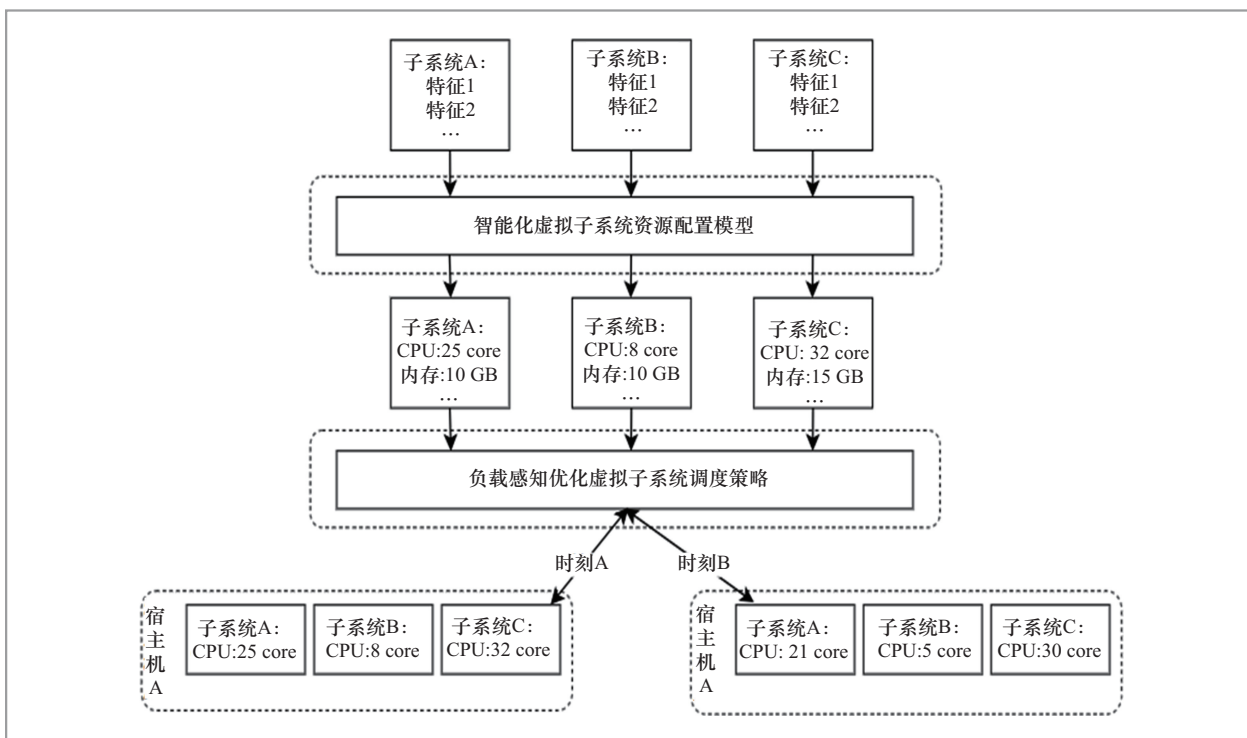


图1 联合优化方法系统架构

(2) 智能化资源配置

将采集到的多维数据输入预先训练好的智能化资源配置模型。该模型基于机器学习算法，能够感知子系统的异构特征，并预测出下一阶段各虚拟化子系统最优的资源需求向量（如表示为<CPU核心数，内存大小，带宽限制>）。此步骤旨在解决资源分配的失衡问题，为后续调度提供科学依据。

(3) 负载感知动态调度

调度模块接收配置模型输出的资源方案，并将其作为初始的资源调度方案。负载感知调度策略采用遗传算法，根据实时负载变化，以最大化资源利用率和保障关键子系统性能为目标，动态调整子系统在物理CPU核心上的分配与执行顺序。需要说明的是，内存资源通过Linux系统的预分配机制已固定，调度算法专注于CPU资源的动态调度。

(4) 配置执行与反馈

遗传算法输出的最优调度方案最终由底层的虚拟化子系统调度器执行。系统的运行状态数据被持续采集，反馈至数据采集模块，形成一个“特征感知-预测配置-动态调度-反馈优化”的闭环，从而实现动态负载的持续自适应优化。

通过上述设计，资源配置模型从宏观上保证了资源分配的合理性，而调度策略则从微观上解决了资源竞争与性能抖动问题，两者协同工作，共同提升了整个虚拟化平台的运行效能。后文第2节和第3节分别对资源配置模型和虚拟化子系统调度策略进行详细描述。

2 应用异构特征感知的资源配置模型

2.1 融合主机典型应用特征分析

融合主机子系统根据其资源消耗模式，

主要划分为以下3类。

- 计算密集型：包括牵引控制、制动控制等实时闭环控制任务，这类任务的CPU利用率长期处于高位，对计算资源高度敏感。

- I/O密集型：包括空调监控、车门控制等周期性感知-执行类任务，这类任务以现场总线或实时以太网I/O为主，网络带宽需求呈周期性峰值。

- 混合型：包括乘客信息系统、车载视频监控等多媒体任务，这类任务既需要视频编解码等计算能力，又依赖高吞吐网络与存储I/O，其资源需求随客流量、列车位置等外部因素动态变化，模式最为复杂。

由此可见，不同子系统的资源消耗模式是不同的，相应地，其最佳资源配置也不同。为确定各子系统的最佳资源配置，本文采用随机森林模型^[15-16]结合二分查找法进行求解。

为了系统性刻画异构虚拟化子系统的特征及其资源分配机制，本文构建了一个多维向量空间模型，用于描述系统特征、资源分配策略与运行状态之间的动态关系，如下所示：

$$\langle T, X, Y \rangle \rightarrow M \quad (1)$$

其中， $T=(t_1, t_2, \dots, t_i)$ 表示子系统的固有特征，特征包括子系统类型、数据量级、线程量级等， $X=(x_1, x_2, \dots, x_i)$ 表示为子系统分配的资源特征，如CPU核数、内存大小， $Y=(y_1, y_2, \dots, y_i)$ 表示子系统在运行过程中检测到的资源消耗指标，如CPU平均利用率、内存平均利用率、平均运行时间， $M=(m_1, m_2, \dots, m_i)$ ， M 中的每一项都是与 Y 对应的，表示将 Y 的各分量作为标签，对应训练得到的随机森林模型集合。

2.2 基于随机森林的资源预测建模

为了得到整体最佳的资源配置，首先需要训练出 M 。本文采用随机森林算法来构建 M 。随机森林是一种集成学习算法，通过构建并组合多个决策树来进行预测，具有不易过拟合、对高维特征数据适应性强等优点，尤其适合处理车载子系统的复杂特征。具体方法为：收集每一个子系统的 T ，然后为每个子系统准备一台对应的虚拟化子系统，并分配 X ，将虚拟化子系统持续运行一段时间，检测到 Y 。收集上述每一个虚拟化子系统的 $[T, X, Y]$ ，汇总后即得到训练数据集。资源预测建模训练数据集示例见表1。

表1为训练数据示例，在训练数据集上使用随机森林模型，训练出 M 模型。训练完成后，输入任意子系统的特征向量 $\langle T, X \rangle$ ，即可通过对应模型 m_i 预测其资源消耗指标 y_i 。

2.3 基于二分查找的资源分配方法

针对各子系统资源分配的最优解问题，本文采用二分查找算法结合 M 评价指标进行求解。具体而言，需要为每个子系统确定其最优的 X 资源分配量，该问题的解决依赖于对系统性能指标的精确量化与评估。

首先，确定各子系统的性能评价指标 y_i 。由于不同子系统呈现差异化的资源消耗特征，其关键性能指标可能涉及CPU利

用率、内存占用率或任务执行时长等不同维度。因此，需要根据各子系统的具体特性及其优化目标，选择最具代表性的性能评价指标 y_i ，将其作为资源分配优化的基准参数。

其次，为 X 和 y_i 分配一个区间。设 $[x_i^{(\min)}, x_i^{(\max)}]$ 为给 x_i 分配的最小值和最大值区间。 $X_d = ([x_1^{(\min)}, x_1^{(\max)}], [x_2^{(\min)}, x_2^{(\max)}], \dots, [x_i^{(\min)}, x_i^{(\max)}])$ ， X_d 为每一个 x_i 的区间合集。 X 的取值必须在 X_d 区间内。设 $[y_i^{(\min)}, y_i^{(\max)}]$ 为 y_i 所期望的一个最小值和最大值的最佳区间，其是本算法对外暴露的用户可配置参数，需要使用者根据实际业务需求和行业经验自行设置。该参数直接影响资源分配结果的质量，具体说明如下。

- 参数性质：期望性能区间是算法的输入参数，而非算法内部自动计算得出的。这种设计允许不同应用场景下的使用者根据自身业务特点灵活设置该参数。

- 设置原则：使用者应基于行业经验和业务需求设置合理的性能区间范围。如果由没有行业经验的人员设置了不合理的性能区间，则最终算法可能无法找到满足条件的配置，或得到的资源分配结果不理想。

若存在一个 X ，使得 y_i 落在 $[y_i^{(\min)}, y_i^{(\max)}]$ 区间，则这个 X 就是一个合理的资源配置。本文使用二分查找法来搜索这个 X 。

以下是通过二分查找法对虚拟化子系统最佳资源分配进行预测的算法流程。

表1 资源预测建模训练数据集示例

子系统固有特征				子系统分配的资源特征		子系统在运行过程中检测到的资源消耗指标		
空调子系统	牵引子系统	数据量级	线程量级	CPU核数/个	内存大小/GB	CPU平均利用率	内存平均利用率	平均耗时/ms
1	0	2 000	1	6	4	0.72	0.34	5
0	1	3 000	2	10	7	0.66	0.57	12

步骤1 按式(2)计算 $X_i^{(\text{mid})}$ 。

$$X_i^{(\text{mid})} = \frac{x_i^{(\text{min})} + x_i^{(\text{max})}}{2} \quad (2)$$

然后, 令 $X^{(\text{mid})} = (X_1^{(\text{mid})}, X_2^{(\text{mid})}, \dots, X_i^{(\text{mid})})$ 。

步骤2 将 $X^{(\text{mid})}$ 作为 \mathbf{X} , 将 $\langle T, X^{(\text{mid})} \rangle$ 给到 m_i , 预测得到 y_i 。

步骤3 若 y_i 落在 $[y_i^{(\text{min})}, y_i^{(\text{max})}]$ 区间, 则 $X^{(\text{mid})}$ 就是要分配的资源配置, 退出算法。

步骤4 若 $y_i < y_i^{(\text{min})}$, 则令 $x_i^{(\text{max})} = X_i^{(\text{mid})} - 1$, 重新执行步骤1~步骤3。

步骤5 若 $y_i > y_i^{(\text{max})}$, 则令 $x_i^{(\text{min})} = X_i^{(\text{mid})} + 1$, 重新执行步骤1~步骤3。

步骤6 若 $y_i^{(\text{max})} < y_i^{(\text{min})}$, 则退出算法, 此时无法分配资源, 可重新设置区间范围等参数, 再执行二分查找法。

3 负载感知优化的虚拟化调度策略

3.1 调度问题分析

上述资源配置模型针对的是单个虚拟化子系统设计, 通过该模型能够为这个子系统确定一个理论上的最佳资源配置方案。然而, 现实情况往往是一台主机上同时运行多个虚拟化子系统, 各子系统对主机资源存在竞争性需求, 且在影响子系统运行效率的各类资源中, 内存等资源不便动态调整, 而能够实时动态调整的主要是CPU等计算资源。因此, 需要通过调度机制动态博弈以获取计算资源份额。为了给所有的子系统进行资源分配优化, 本文采用遗传算法来构建动态的资源调度策略。遗传算法^[17]是一种经典启发式算法, 该算法通过模拟生物进化中的选择、交叉与变异机制, 迭代优化调度方案, 适合本文所面临的问题。

3.2 目标函数

资源配置模型通过随机森林预测了CPU、内存等多种资源指标, 为各子系统提供了初始资源配置方案。然而, 在动态调度阶段, 内存资源通过预分配机制已固定, 无法进行实时调整。因此, 本调度算法专注于CPU资源的动态调度, 这是因为CPU是影响系统运行效率的主要瓶颈资源。实时调整各子系统在物理CPU核心上的分配, 能够有效提升系统整体资源利用率和性能表现。

为实现多子系统间的动态资源调度, 本文提出一种基于遗传算法的优化方法。该方法首先构建一个综合考虑系统性能指标与资源利用率的多目标优化函数, 继而通过遗传算法对该目标函数进行迭代优化, 从而实现对系统资源的动态分配与调度。目标函数的总体形式为:

$$f = ku + (1 - k)p \quad (3)$$

其中, u 表示主机所有物理CPU在指定时间内的平均利用率, 用来评估指定时间内CPU资源的使用情况, p 为所有子系统运行时间的综合参数, 用来评估所有子系统的运行性能。 p 的具体计算方式为:

$$p = \sum_{i=1}^n w_i \frac{a_i}{b_i} \quad (4)$$

其中, n 表示子系统总数, a_i 表示子系统 i 单独运行时的运行时间, b_i 表示子系统 i 在 n 个子系统同时运行时的实际运行时间。对于长时间运行的子系统, 其运行时间定义为周期性运行任务的资源占用的波峰或波谷之间的时间间隔。 w_i 为子系统 i 的权重参数, 代表其重要性和优先级。 w_i 应满足如 $\sum_{i=1}^n w_i = 1$ 的约束条件。如无特殊考虑, 默认 $w_i = \frac{1}{n}$ 。

式 (3) 中的 k 为平衡 u 和 p 的权重参数, 通过调整 k 来调整目标函数 f 在性能和资源利用率之间的侧重点, 以实现调度策略的灵活性。

3.3 遗传算法

本文在传统遗传算法的基础上, 针对 CPU 调度场景进行了适应性修改, 通过交叉、变异、位移等操作, 实现了对各子系统在物理 CPU 核心分配方案上的动态优化。

3.3.1 问题描述

假设有 n 个子系统, z_i 为第 i 个子系统在当前轮次的解。 $Z = \{z_1, z_2, \dots, z_i\}$, Z 为资源调度策略在当前轮次的调度方案。 f 为当前轮次的目标函数值。当存在两个子系统, 有 5 个 CPU 核数时, 解的格式示例见表 2。

表 2 中采用二进制来表示子系统是否使用对应的 CPU 核, 0 代表不使用, 1 代表使用。

假设 g_1 是历史最优的调度方案, g_2 是历史第二好的调度方案。采用维护两个历史最优解 (g_1 和 g_2) 的策略, 其核心设计原则是确保每轮迭代能够产生并更新历史最优解, 从而保证算法持续向更优解方向收敛。从理论角度而言, 维护历史最优解的数量并非算法的关键约束条件, 算法的有效性主要依赖于以下两个核心机制。

- 历史最优解的更新机制: 每轮迭代中, 算法都会将当前轮次的解与历史最优解进行比较, 确保历史最优解持续更新, 从而保证算法单调非退化。

- 解空间的充分探索: 通过遗传操作 (交叉、变异、位移) 对解空间进行系统性探索, 避免算法过早收敛到局部最优。

表 2 解的格式示例

子系统编号	解
1	00111
2	01011

具体做法为: 首先, 可以按照资源配置模型, 确定第一轮初始化的 Z ; 然后, 按照 “计算 $f \rightarrow$ 更新 $[g_1, g_2] \rightarrow$ 生成下一轮的 Z ” 的流程来动态地调度资源分配方案 Z 。

3.3.2 计算 f

按照 Z 给子系统分配 CPU 核数, 然后将子系统持续运行一段时间后, 就可以按照式 (3) 来计算当前轮次的目标函数值 f 。

3.3.3 更新 $[g_1, g_2]$

当第二轮结束时, 将第一轮的 Z 和第二轮的 Z 放在一起, 依据各自的目标函数值 f 进行比较, 确定 $[g_1, g_2]$ 。从第三轮开始, 将当前轮次的 Z 和 g_1 、 g_2 放在一起, 依据各自的目标函数值 f 进行比较, 重新确定 $[g_1, g_2]$ 。

3.3.4 生成下一轮的 Z

依次按照 “交叉、变异、位移、全 0 异常处理” 的流程来生成下一轮的 Z 。该流程中间产生的解称为 Z^* 。执行完上述流程后, 将 Z^* 作为下一轮的 Z 。

(1) 交叉

从第三轮开始才执行交叉操作。提取第 i 个子系统在 g_1 和 g_2 的调度解, 解为一串二进制, 对解中的每一位, 随机选择该位的二进制是 g_1 中的二进制还是 g_2 中的二进制。对每个子系统的每一位都如此操作,

由此产生 Z^* 。对于第一轮和第二轮，直接将 Z 作为 Z^* 。其中交叉操作在CPU调度中的实际含义为：将历史最优解 g_1 和历史第二好解 g_2 的CPU分配方案进行组合，探索两种优秀方案之间的解空间，寻找可能更优的CPU核心分配策略。

(2) 变异

设变异概率为 p_m 。将第 i 个子系统在 Z^* 中的解取出来，对于解中的每一位，以 p_m 的概率将该位的二进制取反，即0变成1，1变成0。对每个子系统的每一位都如此操作，由此更新 Z^* 。若某个子系统变异操作后，解是全0二进制，则不执行该子系统的变异操作。其中变异操作在CPU调度中的实际含义为：以概率 p_m 随机改变某个子系统对某个CPU核心的占用状态（0变1或1变0），用于跳出当前解的邻域，探索新的分配可能性。

(3) 位移

将第 i 个子系统在 Z^* 中的解取出来，对于解中每一个取值为1的位，执行如下操作：先向左移动，若该位不是解的第一位，且前一位取值为0，则以0.25的概率将该位上的1移动到左一位；接着向右移动，若该位不是解的最后一位，且该位后一位取值为0，且该位并未执行向左移动的操作，则以0.25的概率将该位上的1移动到后一位。对每一个子系统都按此方式操作，由此更新 Z^* 。其中，位移操作在CPU调度中的实际含义为：将某个子系统已占用的CPU核心（值为1的位）移动到相邻的未占用核心（值为0的位），这种操作保持了资源分配的总量不变，但改变了分配的位置，有助于在保持当前解质量的同时进行局部优化。

(4) 全0异常处理

若第 i 个子系统在 Z^* 中的解是全0二进制，则随机选择一位，将该位变成1。对

每一个子系统都按此方式操作，由此更新 Z^* 。位移概率设定为0.25，意味着对于每个已占用的CPU核心，有75%的概率保持不动，25%的概率尝试移动到相邻位置。这种设计体现了“大概率保持稳定，小概率进行微调”的调度策略，符合实际调度场景中“不频繁调整，适度优化”的需求。需要说明的是，0.25这个具体的概率值并不是算法的关键参数。算法的核心在于，通过交叉、变异、位移的组合操作，实现遗传进化的效果。只要这些操作能够有效引导算法向更优解收敛，具体的概率参数可以根据实际场景进行调整。

4 实验与结果

为验证所提联合优化方法的有效性，本文搭建了模拟仿真环境，并设计了两组实验，分别对资源配置模型和调度策略的性能进行评估。

4.1 实验环境与设置

受限于列车融合主机真实场景的特殊硬件资源及场所，难以直接进行相应功能测试。为保证实验的可行性与可重复性，本文通过模拟车载融合主机的实验环境对虚拟化子系统资源分配与调度过程进行了相关实验^[18-22]。

硬件平台：服务器搭载Intel® Xeon® Silver 4114 CPU(10核，2.2 GHz)，14 GB DDR4内存。

软件平台：宿主机操作系统为CentOS 7，通过Linux底层资源隔离技术CGroup实现对每个虚拟化子系统的资源实时精细化控制。

模拟子系统负载：为模拟列车融合主

机子系统的真实工况，部署了3个虚拟化子系统实例，分别模拟空调控制子系统、牵引控制子系统和制动控制子系统这3类典型子系统。每个虚拟化子系统被分配固定的CPU核数与内存容量等资源配置。负载生成采用压力测试工具（如 stress-ng^[23]），监控与评估指标包括CPU利用率、内存占用率、任务完成时间等。

4.2 资源配置模型实验

本实验旨在验证本文提出的基于随机森林建模和二分查找法的资源配置模型的预测效果。

4.2.1 实验数据集与特征分析

实验基于内部虚拟化场景数据集，共包含320条样本。数据集模拟了融合主机三大核心子系统在不同资源配置下的运行表现，主要特征如下。

(1) 子系统类型特征 (A_0, A_1, A_2): 采用独热编码表示融合主机子系统类型，其中 $A_0=1$ 表示空调控制子系统， $A_1=1$ 表示牵引控制子系统， $A_2=1$ 表示制动控制子系统。三大子系统分别承担着融合主机环境控制、动力驱动和安全制动的关键功能。

(2) 应用的固有特征 T : 如应用类型、数据量级、线程量级。

(3) 应用分配的资源特征 X : 如CPU、内存。

(4) 应用在运行时检测到的资源消耗指标 Y : 如CPU平均利用率、内存平均利用率、平均运行时间。

4.2.2 模型训练与评估方法

基于随机森林建模和二分查找法扩展实现的训练框架，实验采用以下方法。

(1) 特征选择：以 $[A_0, A_1, A_2, x_1, x_2, x_3, x_4, x_5]$ 为输入特征向量，涵盖子系统类型、负载特征和资源配置。

(2) 多目标预测：针对每个性能指标 ($y_1 \sim y_2$) 分别训练独立的随机森林模型。

(3) 随机森林：100棵决策树，最大深度为4。

(4) 数据划分：采用7:3比例将数据集划分为训练集（224个样本）和测试集（96个样本），以确保模型泛化能力评估的可靠性。

(5) 评价指标：采用平均绝对误差 (mean absolute error, MAE)、均方根误差 (root mean square error, RMSE) 和决定系数 (coefficient of determination, R^2) 作为评价指标，评估预测精度。

4.2.3 预测精度评估结果

预测精度评估结果见表3。由表3可知，CPU平均占用率 (y_1) 和最高占用率 (y_2) 的MAE分别为0.094和0.096， R^2 分别达到0.638和0.623，能够为子系统CPU资源需求评估提供可靠的量化参考；平均内存占用 (y_3) 和最高内存占用 (y_4) 的MAE均为0.002，RMSE均为0.003，有助于配置各子系统的内存资源；平均耗时 (y_5) 的MAE为2.208， R^2 为0.576，

表3 预测精度评估结果

预测指标	物理含义	MAE	RMSE	R^2
y_1	CPU平均占用率	0.094	0.143	0.638
y_2	CPU最高占用率	0.096	0.144	0.623
y_3	平均内存占用	0.002	0.003	0.284
y_4	最高内存占用	0.002	0.003	0.332
y_5	平均耗时	2.208	5.187	0.576

RMSE为5.187,在复杂时序性能预测中展现出了一定的拟合能力,为子系统调度优化提供了时间基准。总体上,随机森林建模和二分查找法在各项性能指标上均表现出较好的预测精度。

4.2.4 资源配置模型自动给出配置方案

实验开始阶段设置了资源配置模型所需的参数,具体如下。

(1) 实验选取资源CPU核数(个)、内存大小(GB)进行分配,3个子系统的CPU核数区间是[[1,10],[1,10],[1,10]],内存区间是[[1,10],[1,10],[1,10]]。

(2) 实验选取的资源消耗指标是“CPU平均占用率”,区间是[[0.95,1.0],[0.93,0.99],[0.92,0.96]]。

该区间设置确保了各子系统在正常运行状态下的性能得以保障,同时避免了资源过度分配。最终得到的资源配置模型自动给出配置方案见表4,为后续调度提供了相较于人工配置更优化的初始方案。

4.3 虚拟化调度策略实验

本文提出的负载感知优化的虚拟化调度策略也称“改进遗传算法”。

4.3.1 实验数据集与特征分析

为模拟融合主机三大典型子系统,本文部署了3个虚拟化子系统实例,分别为

空调控制子系统、牵引控制子系统、制动控制子系统。

改进遗传算法的参数设置如下。

(1) CPU的初始调度方案:基于业务特征的预设分配方案为[111111111,111111111,111111111]。

(2) 变异概率: $p_m=0.1$ (位取反操作)。

(3) 权重因子: $k=0.6$ (平衡利用率与性能)。

(4) 最大轮次:152轮。

每轮实验持续运行,通过实时监控采集CPU利用率、内存占用率、任务响应时间等关键指标。

4.3.2 CPU资源分配策略分析

本实验开始阶段为各子系统设置了资源配置模型所推荐的单体子系统最优CPU配置[8,8,8],然后通过运行改进遗传算法,实时得出在同一个物理服务器上综合了性能及CPU利用率的情况下,各子系统所需的CPU配置。子系统CPU资源分配调度演变见表5。

从第1轮至第152轮的长期监控数据可以看出,第57轮属于这一次运行中的最优配置推荐。

(1) 空调控制子系统:CPU核心分配从初始的8核逐步优化至2~4核,体现了该子系统具有相对稳定的负载特征,算法

表4 资源配置模型自动给出配置方案

子系统固有特征参数					子系统配置方案	
空调	牵引	制动	数据量级	线程量级	CPU核数/个	内存大小/GB
1	0	0	2 000	1	2	3
0	1	0	3 000	1	5	5
0	0	1	3 000	1	5	3

通过适度的资源分配既保证了环境控制功能，又避免了资源浪费。分配策略体现了环境控制系统负载相对可预测的特点。

(2) 牵引控制子系统：分配核心数在1~2核间动态调整，反映了牵引系统随融合主机运行工况变化的特点，在启动、加速和高速运行等不同阶段对计算资源需求存在一定程度的差异。

(3) 制动控制子系统：始终保持较高的资源分配优先级（3~4核），作为安全关键型子系统，制动子系统需要确保紧急情况下的快速响应能力，算法的分配策略体现对安全性的重视。

(4) 最优配置：随着轮次的迭代， f 趋于拟合稳定，说明调度算法是合理有效的，第57轮的目标函数值 f 为本次实验中的最优值，其对应的CPU调度方案为[010000000, 000100000, 001000101]，即CPU核数分配为[1, 1, 3]。从CPU调度方案可看出，有4个CPU核没有被用到，即节约了4个CPU核数资源，这意味着改进算法在满足业务需求的同时，节约了约44%的CPU资源。

(5) 系统特性：所提方法在152轮连续运行中表现出稳定的收敛特性，目标函数值从初始的0.08提升至0.41，并在第

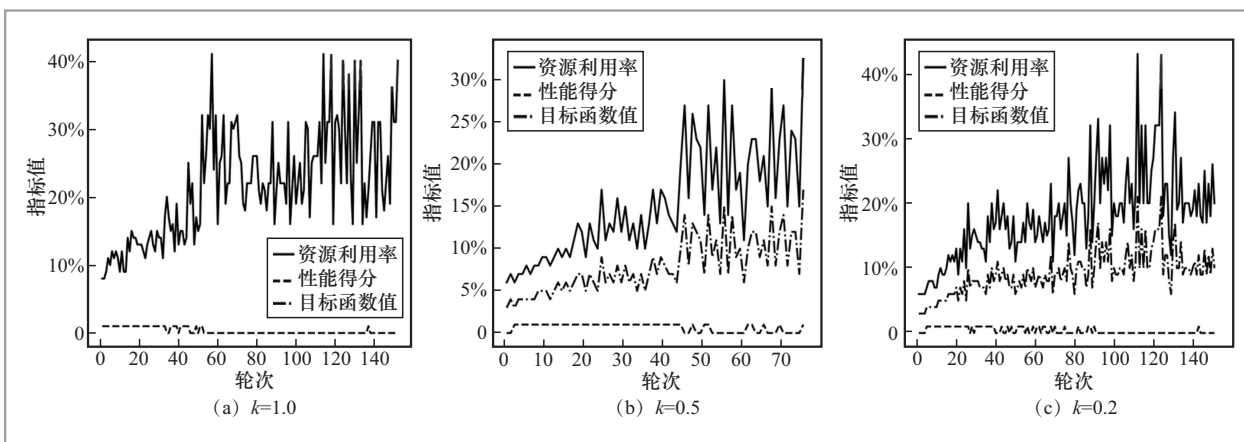
表5 子系统CPU资源分配调度演变

轮次	各类别对应的CPU核数/个			目标函数值 f
	空调	牵引	制动	
0	8	8	8	0.08
1	6	7	6	0.07
2	5	4	5	0.08
3	4	4	5	0.08
...
57	1	1	3	0.41
...
152	2	2	4	0.40

57轮后趋于稳定，说明改进遗传算法能够根据不同权重因子实现差异化的资源管理目标，为融合主机在不同运行状态下的资源优化提供了灵活的解决方案。

4.3.3 权重因子敏感性分析

为验证权重因子 k 对调度策略的影响，本文在相同实验环境下测试了不同 k 值的优化效果，不同 k 值下的性能指标趋势如图2所示。

图2 不同 k 值下的性能指标趋势

(1) $k=0.2$: 侧重性能保障, 性能指标 p_m 会逐渐波动收敛, 适合安全关键场景。

(2) $k=0.5$: 平衡配置, 资源利用率与性能指标 p_m 相互呼应, 注重综合性能。

(3) $k=1.0$: 侧重资源利用, 资源利用率与 f 基本重叠, 但性能指标 p_m 在 0 附近。

该特性验证了本文所提调度策略的灵活性, 运营者可根据融合主机的运行状态(如平稳运行状态与紧急制动状态的对比情况)灵活调整权重, 实现差异化的资源管理目标。

4.3.4 与基线方法对比

本文所提调度策略与传统调度策略的性能对比见表6。由表6可知, 改进遗传算法的CPU平均利用率为22.07%, 相较DRF算法(13.5%)提升幅度达63.5%; 改进遗传算法的目标函数值(0.204)是DRF算法(0.035)的5.83倍。可见, 改进遗传算法在CPU利用率、性能指标及动态负载适应性上均优于DRF算法, 更适合高资源利用率与动态调度场景。

5 结束语

本文针对以列车融合主机为典型应用场景的边缘计算平台虚拟化资源管理中存在的配置不准、调度不优的问题, 提出了一种基于智能化配置与负载感知调度的联合优化方法。本文的主要研究成果如下。

(1) 智能化资源配置模型: 提出了一种基于随机森林建模和二分查找法的资源配置预测模型, 通过深度挖掘融合主机子系统的异构特征, 实现了对虚拟化子系统资源的精准预测。在320个样本的数据集上, CPU占用率预测的MAE差为0.095, R^2 达到0.638, 为资源合理分配提供了科学依据。

(2) 改进遗传调度策略: 设计了一种基于负载感知优化的虚拟化调度策略的改进遗传算法, 区别于传统的基于种群的遗传算法, 该算法只维护两个历史最优解进行演化操作, 具有计算复杂度低、收敛速度快的特点, 通过引入兼顾资源利用率与子系统性能的多目标适应度函数, 解决了混合负载下资源竞争的问题。

未来的研究工作可从以下几个方面展开: 首先, 将当前针对单一主机或节点的优化模型扩展至分布式多节点边缘集群环境, 重点研究跨节点的协同资源管理与容灾机制, 以提升系统整体的可扩展性与可靠性; 其次, 引入深度强化学习等更先进的自适应算法, 利用其强大的在线学习与决策能力应对边缘环境中极致的动态性和不确定性, 进一步提升系统在复杂动态负载下的自主决策与自适应优化能力; 最后, 探索将该方法与云-边-端协同的边缘计算架构进一步融合, 构建层次化的资源管理范式, 并在智能制造、智慧城市及云游戏等更多高动态边缘场景中验证其普适性, 为构建高效、可靠的智能边缘系统提供通用技术支撑。

表6 本文所提调度策略与传统调度策略的性能对比

调度策略	CPU平均利用率	CPU最高利用率	性能指标 p	目标函数值 f	收敛轮数/轮
改进遗传算法	22.07%	43.0%	0.0024	0.204	57
DRF算法	13.5%	16.0%	0.0015	0.035	-

参考文献:

- [1] 李培根. 中国制造2025[J]. 广东科技, 2016, 25(17): 16-19.
Li P G. China manufacturing 2025[J]. Guangdong Science and Technology, 2016, 25(17): 16-19.
- [2] 宋尧, 宋平, 高巍, 等. 面向广域分布式智能计算的运行时算力网络资源协同调度方法研究[J]. 大数据, 2025, 11(3): 3-16.
Song Y, Song P, Gao W, et al. Runtime collaborative scheduling of resources in computing power network oriented to the wide-area distributed intelligent computing environment[J]. Big Data Research, 2025, 11(3): 3-16.
- [3] 张旭东, 何劲涛, 张涛, 等. 基于虚拟化方法的车辆架构测试技术研究[J]. 北京理工大学学报, 2025, 45(8): 815-823.
Zhang X D, He J T, Zhang T, et al. Research on vehicle architecture testing technology based on virtualization method[J]. Journal of Beijing Institute of Technology, 2025, 45(8): 815-823.
- [4] 韩德志, 李楠楠, 毕坤. 云环境下的虚拟化技术探析[J]. 华中科技大学学报(自然科学版), 2012, 40(S1): 262-265.
Han D Z, Li N N, Bi K. Study of virtualization technology in cloud environment[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2012, 40(S1): 262-265.
- [5] Zhang S, Zhang S F, Chen X B, et al. Cloud computing research and development trend[C]//Proceedings of the 2010 Second International Conference on Future Networks. Piscataway: IEEE Press, 2010: 93-97.
- [6] 杜振环, 马志明, 王延超, 等. 基于虚拟化的列车融合控制技术研究[J]. 2025(3): 12-15.
Du Z H, Ma Z M, Wang Y C, et al. Research on virtualization based train fusion control technology [J]. Railway locomotive and Motor Car, 2025(3): 12-15.
- [7] Kim C, Choi S, Huh J. GVTS: global virtual time fair scheduling to support strict fairness on many cores[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(1): 79-92.
- [8] 徐伟. 基于云平台的城轨列车自动监控系统总体架构研究[J]. 铁道通信信号, 2024, 60(9): 73-80.
Xu W. Research on the architecture of automatic train supervision system for urban rail transit based on cloud platforms[J]. Railway Signalling & Communication, 2024, 60(9): 73-80.
- [9] 时光, 郭玉东, 王晓睿, 等. Xen中VCPU调度算法分析[J]. 计算机工程与设计, 2010, 31(18): 4116-4119, 4123.
Shi G, Guo Y D, Wang X R, et al. Analysis of VCPU scheduling algorithm in Xen[J]. Computer Engineering and Design, 2010, 31(18): 4116-4119, 4123.
- [10] Xiao Z, Song W J, Chen Q. Dynamic resource allocation using virtual machines for cloud computing environment[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1107-1117.
- [11] Ghodsi A, Zaharia M, Hindman B, et al. Dominant resource fairness: fair allocation of multiple resource types[J]. USENIX Association, 2011: 323-336.
- [12] Mishra M, Das A, Kulkarni P, et al. Dynamic resource management using virtual machine migrations[J]. IEEE Communications Magazine, 2012, 50(9): 34-40.
- [13] Wang W, Liang B, Li B C. Multi-resource fair allocation in heterogeneous cloud computing systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2015, 26(10): 2822-2835.
- [14] Lan T, Kao D, Chiang M, et al. An axiomatic theory of fairness in network resource allocation[C]//2010 Proceedings IEEE INFOCOM. Piscataway: IEEE Press, 2010: 1-9.
- [15] 何明卫, 纳静笠, 朱权, 等. 城市小汽车用户公交使用的影响因素分析[J]. 重庆交通大学学报(自然科学版), 2024, 43(5): 99-105.
He M W, Na J L, Zhu Q, et al. Influenc-

- ing factors on the use of bus by urban car users[J]. Journal of Chongqing Jiaotong University(Natural Science), 2024, 43(5): 99-105.
- [16] Breiman L. Random forests[J]. Machine Learning, 2001, 45(1): 5-32.
- [17] 黄鹏飞, 谈方娇, 王浩, 等. 基于改进免疫遗传算法的海铁转运设备作业调度优化研究[J]. 重庆交通大学学报(自然科学版), 2025(6): 97-107.
Huang P F, Tan F J, Wang H, et al. Scheduling optimization for sea-rail transport equipment based on improved immune genetic algorithm[J]. Journal of Chongqing Jiaotong University (Natural Science), 2025(6): 97-107.
- [18] 陈青云. 智慧城市轨道交通云原生探讨[J]. 城市轨道交通研究, 2023, 26(6): 12-16.
Chen Q Y. Discussion on cloud-native for smart city rail transit[J]. Urban Rail Transit Research, 2023, 26(6):12-16.
- [19] 朱涛, 陈栋, 张斌, 等. 铁路运输调度业务中台的研究与实践[J]. 智慧轨道交通, 2024(5): 91-94.
Zhu T, Chen D, Zhang B, et al. Research and Practice of Railway Transportation Dispatching Business Middle-ware[J]. Smart Rail Transit, 2024(5): 91-94.
- [20] Xavier M G, Neves M V, Rossi F D, et al. Performance evaluation of container-based virtualization for high performance computing environments[C]// Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. Piscataway: IEEE Press, 2013: 233-240.
- [21] Scheepers M J. Virtualization and containerization of application infrastructure: a comparison[C]//Proceedings of the 21st Twente Student Conference on IT. Enschede: University of Twente, 2014: 1-7.
- [22] Sturley H, Fournier A, Salcedo-Navarro A, et al. Virtualization vs. containerization, a comparative approach for application deployment in the computing continuum focused on the edge[J]. Future Internet, 2024, 16(11): 427.
- [23] 杨晓伟, 廖光灯, 胡越明. 基于Xen的X86虚拟机性能调优[J]. 计算机工程, 2006, 32(24): 251-253.
Yang X W, Liao G D, Hu Y M. Performance tuning of Xen-based X86 virtual machine[J]. Computer Engineering, 2006, 32(24): 251-253.

作者简介



齐玉玲 (1985-), 女, 中车南京浦镇车辆有限公司副高级工程师, 主要研究方向为列车电气系统设计、列车网络控制技术、车辆网络安全。



黄涛 (1981-), 男, 中车南京浦镇车辆有限公司正高级工程师, 主要研究方向为列车网络控制技术、车辆网络安全。



刘国菲（1980-），女，中车南京浦镇车辆有限公司高级工程师，主要研究方向为轨道交通列车通信系统的设计。



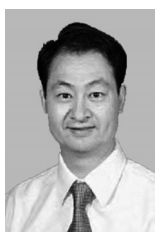
张军贤（1982-），女，中车南京浦镇车辆有限公司正高级工程师，主要研究方向为轨道车辆网络系统、智能化系统及车辆网络安全。



鲍春晓（1993-），男，江苏鸿程大数据技术与应用研究院有限公司高级工程师，主要研究方向为自然语言处理算法。



吴江鹏（1995-），男，江苏鸿程大数据技术与应用研究院有限公司高级工程师，主要研究方向为大数据云原生。



黄宜华（1962-），男，计算机软件新技术国家重点实验室（南京大学）教授、博士生导师，主要研究方向为大数据处理与并行计算。

收稿日期: 2025-10-23

通信作者: 吴江鹏, wjp_199502@163.com

基金项目: 江苏省前沿技术研发计划(No.BF2024005)

Foundation Item: Jiangsu Provincial Frontier Technology Research and Development Program (No.BF2024005)