

物联网背景下动态权重负载均衡算法设计

邱秀荣¹ 杨瑞丽¹

QIU Xiurong YANG Ruili

摘要

针对物联网背景下终端设备类型复杂、运行状态动态变化显著及服务需求多样化等问题,文章设计了一种动态权重负载均衡算法。该算法引入改进型哈希环结构与渐进式会话转移机制,并通过设备状态和服务资源指标,构建动态权重评估模型。实验表明,在20 000个节点规模的实验环境中,该算法的平均服务响应时间与传统负载均衡算法相比均有所提高;同时故障恢复时间控制在9 s以内,在一定程度上提高了资源利用率。

关键词

物联网; 负载均衡; 动态权重

doi: 10.3969/j.issn.1672-9528.2025.10.018

0 引言

随着物联网(internet of things, IoT)技术的快速发展,全球联网设备规模呈指数级增长^[1]。据预测,到2025年全球活跃物联网设备将突破400亿台,由此产生的异构终端设备类型复杂(如传感器、边缘计算节点等)、运行状态动态波动显著(如设备在线率变化、资源占用率浮动)以及服务需求多样化(实时监控、数据聚合、事件驱动响应等)等特征,对传统负载均衡技术提出了严峻挑战^[2]。在此背景下,如何设计适应物联网特性的负载均衡算法,已成为提升系统性能、可靠性和资源利用率的热点问题。

现有算法存在显著局限性,轮询与加权轮询适用于服务器同构场景,但无法处理动态负载^[3];最少连接数在长连接场景中易忽视服务器处理能力差异^[4];地理路由算法基于设备位置选择边缘节点,但未考虑节点负载^[5];能效优化算法通过休眠机制降低能耗,但牺牲实时性^[6]。

综上所述,现有算法多关注单一指标(延迟或能耗),缺乏多维动态融合与高效容错机制,难以满足IoT场景的综合需求。因此本文提出动态权重负载均衡算法(dynamic weighted load balancing algorithm, DW-LB),通过实时整合CPU、内存、延迟、电量四类指标,构建动态权重评估体系,以适应复杂的物联网环境。

1. 商丘工学院信息与电子工程学院 河南商丘 476000

[基金项目] 河南省高等学校重点科研项目“基于智能物联网的负载均衡关键技术研究”(24B413009);商丘市科技攻关项目“物联网环境下节点负载均衡策略研究”(2024029);商丘工学院科研基金资助项目

1 动态权重负载均衡算法设计

1.1 动态权重模型设计

通过对文献[7-9]的分析,结合其优点,对动态权重模型进行设计。定义服务器节点*i*在时间*t*的权重公式为:

$$W_i(t) = \underbrace{\alpha \cdot e^{-\lambda C_i(t)}}_{\text{CPU效用}} + \underbrace{\beta \cdot \max(1 - \frac{M_i(t)}{M_{th}}, 0)}_{\text{内存效用}} + \underbrace{\gamma \cdot \frac{1}{L_i(t) + e}}_{\text{延迟效用}} + \underbrace{\delta \cdot \frac{1}{1 + e^{-k(B_j(t) - B_{th})}}}_{\text{电量效用}} \quad (1)$$

式中: α 、 β 、 γ 、 δ 分别为CPU、内存、延迟、电量的权重系数,通过熵权法动态调整; $B_j(t)$ 中, j 为终端设备编号,与服务器节点*i*形成映射关系。公式中各参数说明如表1所示。

表1 公式中参数说明

符号	含义	单位/范围
$W_i(t)$	服务器节点 <i>i</i> 在时间 <i>t</i> 的权重	无单位(归一化值)
α 、 β 、 γ 、 δ	权重系数 (CPU、内存、延迟、电量)	动态调整
$C_i(t)$	浮点数	设备电量阈值(默认20%)
$M_i(t)$	服务器节点 <i>i</i> 的CPU利用率	百分比(0%~100%)
$L_i(t)$	服务器节点 <i>i</i> 的内存占用率	百分比(0%~100%)
$B_j(t)$	终端设备 <i>j</i> 的剩余电量	百分比(0%~100%)
λ	CPU衰减因子 (防止过载)	默认0.05
M_{th}	内存占用率阈值	默认80%
e	平滑因子(避免除零错误)	默认0.1
k	电量敏感性参数 (控制权重变化速率)	默认0.1
B_{th}	设备电量阈值	默认20%

1.2 权重系数动态调整

相比人工设定权重，熵权法^[10]具有两大优势：一方面通过指标变异程度自动分配权重，避免人为偏见；另一方面是当某指标（如 CPU）波动加剧时，其熵权自动提升。因此通过熵权法动态计算 α 、 β 、 γ 、 δ ，可消除人工设定权重的主观性，并自动适应指标波动。具体步骤如下：

（1）构建标准化指标矩阵 $X = [x_{ij}]_{m \times n}$ ，其中 x_{ij} 为节点 i 的指标 j 的归一化值；

（2）计算各指标熵值，计算公式为：

$$E_j = -\frac{1}{\ln m} \sum_{i=1}^m p_{ij} \ln p_{ij} \quad (2)$$

式中： $p_{ij} = \frac{x_{ij}}{\sum x_{ij}}$ 。

（3）权重公式为：

$$w_j = \frac{1 - E_j}{\sum_{k=1}^n (1 - E_k)} \quad (3)$$

熵权法通过量化指标的动态分配权重，适用于物联网场景中高动态的负载变化。其优势在于自动化权重调整，避免人工预设的主观性。然而，当指标发生瞬时突变（如 CPU 利用率从 10% 骤升至 90%）时，熵权法的计算周期（60 s）可能导致权重更新延迟。未来可通过滑动窗口机制（如每 10 s 局部更新）优化实时性。

2 DW-LB 算法设计

2.1 算法设计步骤

动态权重负载均衡算法（DW-LB）包含监控采集、权重计算、请求分配、故障迁移四大模块，具体步骤如下：

步骤 1：初始化参数

（1）服务器节点列表初始化，注册所有边缘节点 IP 与基础配置。

（2）权重系数初始化，初始值根据历史数据预估， $\alpha=0.3$ 、 $\beta=0.3$ 、 $\gamma=0.2$ 、 $\delta=0.2$ 。

（3）启动轻量级监控代理，部署于服务器与设备端，开启 UDP 数据上报。

步骤 2：实时数据采集与预处理

（1）服务器端，每秒采集 CPU、内存、网络延迟等数据。

（2）设备终端，每 5 s 上报剩余电量、信号强度、GPS 坐标。

步骤 3：动态权重计算

（1）利用熵权法更新系数，每 60 s 根据最新监控数据重新计算 α 、 β 、 γ 、 δ 。

（2）权重计算，按式（3）计算各节点权重，再利用 Softmax 归一化后使指数平滑。

步骤 4：请求分配策略

（1）加权概率选择，对设备请求 Req_j ，按归一化权重随机选择目标节点。

（2）优先级调整，对实时性敏感请求（如控制指令），优先选择延迟效用高的节点。其中，实时性请求判定标准如下：

① MQTT QoS=1 的控制指令；

② 延迟敏感型请求（ $L_{req} < 50 \text{ ms}$ ）；

③ 工业心跳包（周期 $< 1 \text{ s}$ ）。

（3）故障检测与迁移。每 2 s 向所有节点发送 ICMP 探测包，连续 3 次超时判定故障。从 Redis 备份节点拉取故障节点最近 30 s 的会话状态（MQTT 连接、设备上上下文）。根据一致性哈希环，将故障虚拟节点映射至下一可用节点。

最后通过 RDMA 技术快速同步内存数据，迁移延迟控制在 50 ms 内。

步骤 5：能效优化机制

（1）设备端节能策略，对电量低于 B_{th} 的设备，降低其请求频率（从 1 Hz 调整至 0.2 Hz）。

（2）服务器休眠调度机制，若某节点连续 5 min 负载低于 10%，关闭冗余虚拟机释放资源。

2.2 算法伪代码框架

该算法采用 Python 语言，算法中的参数如表 2 所示。

表 2 算法中参数说明

变量	类型	含义
nodes	列表	服务器节点
weights	字典	节点动态权重（ α 、 β 、 γ 、 δ ）
B_{th}	浮点数	设备电量阈值（默认 20%）

结合前面对算法步骤的分析，该算法的伪代码为：

输入：服务器节点列表 nodes，设备请求队列 requests

输出：请求分配结果 allocations

def DW-LB(nodes, requests):

 # 1. 初始化监控代理

 monitor = UDPMonitor(nodes)

 while True:

 # 2. 实时数据采集（CPU/ 内存 / 延迟 / 电量）

 metrics = monitor.collect_metrics()

 # 3. 动态权重计算（熵权法）

 weights = entropy_weight(metrics)

 normalized_weights = softmax(weights)

 # 4. 请求分配

 for req in requests:

 if req.is_realtime(): # 实时性敏感请求

 target = select_min_latency_node(nodes, normalized_weights)

 else: # 普通请求

 target = weighted_random_select(nodes, normalized_weights)

```

        allocations.add(req, target)
# 5. 故障检测与迁移
failed_nodes = detect_failures(nodes)
migrate_sessions(failed_nodes)

return allocations

```

通过上述对 DW-LB 算法进行设计，可得该算法的时间复杂度主要来自熵权法计算和请求分配，设节点数为 m ，指标数为 n ，则该算法的时间复杂度为 $O(mn)$ ，而空间复杂度为 $O(mn)$ ，用于存储监控数据。实验表明，在 20 000 个节点规模下，单次权重计算耗时 <15 ms，满足实时性需求。

3 实验与结果分析

3.1 实验环境

模拟搭建 20 000 个 IoT 设备、4 台服务器的测试平台，其中 1 个 32 核 /64 GB，3 个 16 核 /32 GB 的服务器。选择轮询算法、加权最少连接数算法作为对比算法，基于上述实验环境，本文三种算法的响应时间、负载均衡度、故障恢复时间为评估指标，对三种算法进行模拟实验。

3.2 实验结果分析

不同设备规模下的平均响应时间如表 3 所示，从表 3 可以看出 DW-LB 算法响应时间曲线在 10 000 个节点时平均响应时间为 43 ms，15 000 个节点时平均响应时间为 47 ms；而轮询算法（Round Robin）从 78 ms 升至 85 ms。最少连接数算法（Least Connections），响应时间介于两者之间，15 000 设备时达到 70 ms，较 DW-LB 高 49%。

表 3 不同设备规模下的平均响应时间

设备规模 / 个	DW-LB /ms	Round Robin /ms	Least Connections /ms
5 000	45	78	65
10 000	43	82	68
15 000	47	85	70
20 000	42	79	63

通过服务器负载标准差指标对比了 DW-LB 与传统算法的负载均衡效果，其负载均衡度对比结果如表 4 所示，标准差越小，表明各服务器节点的负载分布越均衡。表中数据为 10 次重复实验的 95% 置信区间，DW-LB 负载标准差 2.8%。熵权法动态调整使 DW-LB 均衡度较轮询提升 6 倍，在一定程度上优化了资源利用率。

表 4 负载均衡度对比结果

算法	负载标准差 /%
DW-LB	2.8
Round Robin	17.1
Least Connections	12.6

在故障模拟测试中，故障测试包含服务器宕机、网络隔离、CPU 过载等，通过对比动态权重算法（DW-LB）与传统轮询算法（Round Robin）的故障恢复时间，DW-LB 将故障恢复效率提升约 63.6%，满足物联网场景对高可用性（99.99%）的严苛要求。故障恢复时间对比如表 5 所示。

表 5 故障恢复时间对比表

算法	故障恢复时间 /s
DW-LB	8.2
Round Robin	32.7
Least Connections	24.5

通过以上实验分析，DW-LB 的稳定低延迟（< 50 ms）满足车联网、工业控制等场景的实时性要求（通常需 < 100 ms）；响应时间不受设备规模显著影响，证明算法适用于设备多样性的物联网部署。

4 结论

本文提出的 DW-LB 算法通过动态权重模型与增量会话迁移机制，有效解决了物联网场景中的负载均衡难题。实验证明其在响应速度、资源利用率与容错性上均优于传统算法。但本算法目前适用于中等规模边缘计算场景（< 50 000 个节点），超大规模部署时存在元数据同步开销问题，下一步将进一步研究基于 LSTM 的权重预测模型。总之，该算法在物联网负载均衡领域具有理论创新性与工程实用性，为边缘计算资源调度提供了可靠解决方案。

参考文献：

- [1] 陈亮, 李峰, 任保全, 等. 软件定义物联网研究综述 [J]. 电子学报, 2021, 49(5): 1019-1032.
- [2] 贺鹏飞, 范鹏飞, 尹千慧, 等. 基于负载均衡算法的 Hyperledger Fabric 共识机制研究 [J]. 计算机工程, 2022, 48(11): 170-176.
- [3] YUDI HADIWANDRA T. Perbandingan algoritma weighted least connection dan weighted round robin pada load balancing berbasis docker swarm [J]. Journal of cloud computing optimization, 2024, 12(3): 45-66.
- [4] VITTAL S, SINGH M K, FRANKLIN A A. Adaptive network slicing with multi-site deployment in 5G core networks [C] // 2020 6th IEEE Conference on Network Softwarization (NetSoft). Piscataway: IEEE, 2020: 227-231.
- [5] SELVARAJ S, DAMODARAN A. An Intelligent Relay Selection with Optimized Cluster-Based Routing Algorithm for Multi-Hop Wireless Sensor Networks [J/OL]. International journal of computer networks and applications (IJCNA), 2025 [2025-09-11]. <https://ijcna.org/Manuscripts/IJCNA-2025-O-04.pdf>.

基于改进 YOLOv11 的驾驶员面部疲劳检测研究

赵治淳¹ 朱立忠¹
ZHAO Zhichun ZHU Lizhong

摘要

由疲劳驾驶导致的交通事故频发,已成为交通安全隐患之一。为减少道路安全隐患,提高交通安全性,文章提出了一种基于改进 YOLOv11n 的驾驶员疲劳检测方法。首先,使用 EfficientNetV2 网络替换 Backbone 网络,在基础网络中嵌入注意力机制,提升了检测精确度与速度;其次,引入 NWD 损失函数(normalized wasserstein distance),提升小目标检测精度;最后,在检测头模块引入了 Detect-ASFF 模块,提升模型在复杂场景下的检测精度。测试结果表明,与原模型相比,改进后的算法平均精度提高了 1.9%,准确度提高了 2.2%,参数量下降了 50%。在 CEW 数据集(closed eyes in the wild)上表现出较高的准确率与实时性、鲁棒性增强。

关键词

疲劳检测; YOLOv11; 疲劳驾驶

doi: 10.3969/j.issn.1672-9528.2025.10.019

0 引言

近年来,疲劳驾驶导致的交通事故逐渐增多,保证驾驶安全成为一个问题。若驾驶员在长时间连续驾驶、睡眠不足或身体疲劳的状态下驾驶车辆,易出现反应迟钝、注意力不集中的情况,从而显著增加交通事故发生的概率。然而,准确地判断不同情况下驾驶员的疲劳程度成为一个问题。年龄、性别及种族不同的驾驶员的面部差异,环境中不同的光照,阴影等因素均可能干扰检测效果。因此,实时地检测驾驶员疲劳程度,开发相应的系统成为疲劳驾驶检测的重要研究方向。

驾驶员疲劳具体表现为眼睛闭合或半闭合,嘴部微张、

头部倾倒及生理信号波动等。通过眼部特征分析、头部姿态分析、表情识别、3D 建模以及生理信息检测等方法判断疲劳程度。生理信号检测虽然精度较高,但依赖接触式传感器,用户体验较差,难以大规模应用。基于行为特征的方法属于间接检测,易受驾驶习惯影响,误报率较高。基于计算机视觉的非接触式疲劳检测成为主流研究方向,其核心是通过人脸特征分析疲劳状态。不同方法中,基于驾驶员面部特征分析检测的速度较快、准确性高、鲁棒性好、适用范围广,是近年来的主流研究方向之一。其中,相比于双阶段方法,单阶段方法中的 YOLO 系列模型算法具有较快的处理速度。因此本文采用单阶段检测方法^[1]。

蔡忠祺等人^[2]在 YOLOv8n 网络中引入 Ghost 卷积、添

1. 沈阳理工大学自动化与电气工程学院 辽宁沈阳 110159

[6] 邵炜世,皮德常,邵仲世.学习驱动的分布式异构混合流水车间批量流能效调度优化[J].控制理论与应用,2024,41(6): 1018-1027.

[7] HARENI M, KRISHNA S, AYYAPPAN S, et al. Sharma dynamic resource allocation with entropy-Based CPU weighting in mobile edge computing[C]//2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). Piscataway: IEEE, 2022: 1892-1905.

[8] XIE Z G, SONG X, CAO J, et al. Energy-aware task scheduling via sigmoid-based battery modeling in distributed systems[J]. ETRI journal, 2022, 44(5): 746-758.

[9] LIU F Z, YU H, HUANG J W, et al. Joint service migration and

resource allocation in edge-IoT system based on deep reinforcement learning[J]. IEEE internet of things journal, 2024, 11(7): 11341-11352.

[10] 陈运,陈新,陈伟建.信息论与编码(第4版)[M].北京:电子工业出版社,2023.

【作者简介】

邱秀荣(1983—),女,河南商丘人,硕士,副教授,研究方向:物联网负载均衡。

杨瑞丽(1990—),女,河南商丘人,硕士,讲师,研究方向:物联网通信技术。

(收稿日期:2025-05-20 修回日期:2025-10-13)