



(12) 发明专利

(10) 授权公告号 CN 119179564 B

(45) 授权公告日 2025.01.28

(21) 申请号 202411690259.3

G06N 3/0442 (2023.01)

(22) 申请日 2024.11.25

G06N 20/10 (2019.01)

(65) 同一申请的已公布的文献号

G16Y 30/00 (2020.01)

申请公布号 CN 119179564 A

(56) 对比文件

CN 111930511 A, 2020.11.13

(43) 申请公布日 2024.12.24

CN 117931390 A, 2024.04.26

(73) 专利权人 四川万物纵横科技股份有限公司

审查员 韩俊樱

地址 610041 四川省成都市高新区天府大道中段1388号12栋7层1号

(72) 发明人 杨帆 刘瑶 孙凯 胡涛

(74) 专利代理机构 成都贞元会专知识产权代理有限公司 51390

专利代理人 韦海英

(51) Int.Cl.

G06F 9/48 (2006.01)

权利要求书4页 说明书13页 附图1页

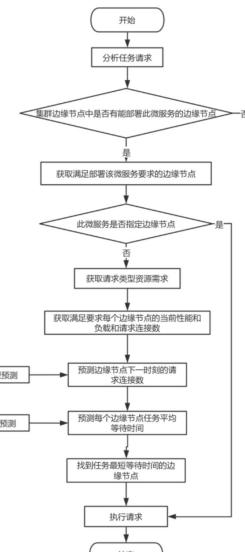
G06F 9/50 (2006.01)

(54) 发明名称

一种基于LSTM-SVM的物联网负载均衡任务调度方法及装置

(57) 摘要

本发明公开了一种基于LSTM-SVM的物联网负载均衡任务调度方法及装置，包括：接收任务请求；获取任务请求的性能指标需求和负载指标需求；获取边缘节点集群和边缘节点的性能指标、负载指标和当前请求连接数；根据当前请求连接数和下一时刻请求连接数基于加权算法计算出边缘节点请求连接数；将边缘节点的性能指标、负载指标和边缘节点请求连接数输入任务等待时间模型，输出预测结果，预测结果包括每个边缘节点的任务等待时间，任务等待时间模型为通过PSO-SVM算法构建的模型；对预测结果进行排序，筛选出所有边缘节点中任务等待时间最短的边缘节点。本发明提高了边缘节点集群吞吐率和响应成功率，还大幅度缩短了并发请求的平均等待时间。



1. 一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,包括:
  - 接收任务请求;
  - 获取所述任务请求的综合性能指标需求和综合负载指标需求;
  - 获取边缘节点集群和边缘节点的综合性能指标、综合负载指标和当前请求连接数;
  - 判断所述任务请求是否指定所述边缘节点集群中的某个边缘节点作为指定边缘节点,若是,则通过所述指定边缘节点部署所述任务请求对应的任务,若否,则进入下一步骤;
  - 根据所述综合性能指标需求和综合负载指标需求设置所述边缘节点的综合性能指标最大值、综合负载指标最大值、综合性能指标阈值、综合负载指标阈值和请求连接数阈值;
  - 判断当前请求连接数是否超过所述请求连接数阈值,若否,则将所述当前请求连接数输入请求连接数预测模型,预测得到所述边缘节点的下一时刻请求连接数,所述请求连接数预测模型为通过LSTM算法构建的模型;
  - 根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数;
  - 判断所述边缘节点请求连接数是否超过所述请求连接数阈值,若否,则判断所述综合性能指标是否超过所述综合性能指标最大值,若否则进入下一步骤;
  - 判断所述综合负载指标是否超过所述综合负载指标最大值,若否则进入下一步骤;
  - 判断所述综合性能指标需求是否超过所述综合性能指标阈值,若否则进入下一步骤;
  - 判断所述综合负载指标需求是否超过所述综合负载指标阈值,若否则进入下一步骤;
  - 将所述边缘节点的综合性能指标、综合负载指标和所述边缘节点请求连接数输入任务等待时间模型,输出预测结果,所述预测结果包括每个所述边缘节点的任务等待时间,所述任务等待时间模型为通过PSO-SVM算法构建的模型;
  - 对所述预测结果进行排序,筛选出所有所述边缘节点中任务等待时间最短的所述边缘节点,得到最佳边缘节点,通过所述最佳边缘节点部署所述任务请求对应的任务。
2. 根据权利要求1所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,获取边缘节点集群,包括:
  - 获取原始边缘节点集群;
  - 在所述原始边缘节点集群中筛选出满足部署条件的所有原始边缘节点形成新边缘节点集群,所述部署条件为判断所述原始边缘节点是否拥有能够部署所述任务请求对应任务硬件和软件资源;
  - 将所述新边缘节点集群作为所述边缘节点集群。
3. 根据权利要求1所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,通过LSTM算法构建所述请求连接数预测模型,包括:
  - 基于LSTM算法构建LSTM模型;
  - 根据所述边缘节点的请求连接数历史数据和具体时刻构建输入矩阵;
  - 对所述输入矩阵进行标准化处理,得到标准化数据;
  - 将所述标准化数据划分为训练集和测试集;
  - 根据所述训练集对所述LSTM模型进行训练,在训练过程中,将ReLU函数作为所述LSTM模型的激活函数,采用Hyperopt对所述LSTM模型进行模型参数调整,采用Dropout方法优化所述LSTM模型的层间过拟合程度,采用Adam方法对所述LSTM模型进行优化,得到所述请求

连接数预测模型。

4. 根据权利要求3所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,对所述输入矩阵进行标准化处理,得到标准化数据,包括:

计算所述输入矩阵的均值;

计算所述输入矩阵的方差;

计算所述输入矩阵与所述均值的差值;

计算所述差值与所述方差的比值,得到所述标准化数据。

5. 根据权利要求1所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,通过PSO-SVM算法构建任务等待时间模型,包括:

基于SVM算法构建SVM模型;

采用PSO算法对SVM模型进行优化;

将所述边缘节点的性能指标历史数据、负载指标历史数据和请求连接数历史数据作为模型输入;

根据所述模型输入对优化后的所述SVM模型进行训练,得到所述任务等待时间模型。

6. 根据权利要求5所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,采用PSO算法对SVM模型进行优化,包括:

初始化粒子群,每个粒子表示一组SVM模型参数,所述粒子的位置表示SVM模型的惩罚系数,所述粒子的速度表示SVM模型的核函数;

更新所述粒子的位置和速度,将预测平均误差值 $g[\gamma(t)]$ 作为确定所述粒子的最优位置的目标函数适应值,对所述SVM模型参数寻优,所述预测平均误差值的计算公式如下:

$$g[\gamma(t)] = \left( \frac{M_{\text{error}}}{M_{\text{total}}} * 100 \right),$$

$M_{\text{error}}$ 为预测的时间数据的误差, $M_{\text{total}}$ 为预测数据的数量;

所述粒子的位置更新公式为:

$$p_i(t+1) = \begin{cases} p_i(t), g[\gamma(t+1)] \geq g[p_i(t)] \\ \gamma(t+1), g[\gamma(t+1)] < g[p_i(t)] \end{cases},$$

将 $P_i(t+1)$ 代入粒子群算法迭代公式:

$$v_{ij}(t+1) = w v_{ij}(t) + c_1 r_{1j}(t)[p_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[p_{gj}(t) - x_{ij}(t)],$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1),$$

$P_i(t)$ 代表粒子*i*在当前时刻的位置, $P_i(t+1)$ 代表粒子*i*在下一时刻的位置, $g[\gamma(t+1)]$ 代表下一时刻的预测平均误差值, $g[P_i(t)]$ 表示粒子*i*当前的适应值, $\gamma(t+1)$ 表示下一时刻的待优化参数,其中w为惯性权重因子, $c_1$ 、 $c_2$ 为加速因子; $r_{1j}(t)$ 、 $r_{2j}(t)$ 为在[0,1]之间相对独立变化的随机函数, $x_{ij}(t)$ 表示t时刻粒子*i*在第j维空间的位置, $x_{ij}(t+1)$ 表示t+1时刻粒子*i*在第j维空间的位置, $v_{ij}(t)$ 表示t时刻粒子*i*在第j维空间的速度,即每次迭代中粒子移动的距离, $v_{ij}(t+1)$ 表示t+1时刻粒子*i*在第j维空间的速度, $P_{ij}(t)$ 为单个粒子*i*在所优化的第j个参数的历史最优解, $P_{gj}(t)$ 表示所有粒子在所优化的第j参数中的历史最优解;

根据所述粒子群算法迭代公式不断迭代调整更新粒子位置和速度,直到达到预设的停

止条件为止,最终得到最优的SVM模型参数。

7.根据权利要求1所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,获取所述边缘节点的综合性能指标、综合负载指标,包括:

  获取所述边缘节点的CPU利用率、硬盘利用率、内存利用率和带宽利用率;

  根据所述边缘节点的CPU利用率、硬盘利用率、内存利用率和带宽利用率基于优序图法计算得到所述边缘节点的综合性能指标;

  获取所述边缘节点的CPU大小、硬盘大小、内存大小和带宽大小;

  根据所述边缘节点的CPU大小、硬盘大小、内存大小和带宽大小基于优序图法得到所述边缘节点的综合负载指标。

8.根据权利要求1所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,还包括对所述预测结果进行评估,对所述预测结果进行评估的步骤包括:

  获取在所述最佳边缘节点部署所述任务后,所述边缘节点集群中各个边缘节点的任务等待时间;

  根据所述边缘节点集群中各个边缘节点的任务等待时间计算所述边缘节点集群的平均任务等待时间;

  判断所述平均任务等待时间是否不超过预设阈值,所述预设阈值为所述边缘节点集群负载均衡下的平均任务等待时间。

9.根据权利要求1所述的一种基于LSTM-SVM的物联网负载均衡任务调度方法,其特征在于,根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数,包括:

  对所述当前请求连接数和所述下一时刻请求连接数进行重要性排序;

  根据排序后所述当前请求连接数和所述下一时刻请求连接数,基于优序图法计算所述当前请求连接数的第一权重和所述下一时刻请求连接数的第二权重;

  根据所述第一权重、第二权重、所述当前请求连接数和所述下一时刻请求连接数加权计算得到所述边缘节点请求连接数。

10.一种基于LSTM-SVM的物联网负载均衡任务调度装置,其特征在于,所述装置包括:

  接收模块,所述接收模块用于接收任务请求;

  第一获取模块,所述第一获取模块用于获取所述任务请求的综合性能指标需求和综合负载指标需求;

  第二获取模块,所述第二获取模块用于获取边缘节点集群和所述边缘节点的综合性能指标、综合负载指标和当前请求连接数;

  第一判断模块,所述第一判断模块用于判断所述任务请求是否指定所述边缘节点集群中的某个边缘节点作为指定边缘节点,若是,则通过所述指定边缘节点部署所述任务请求对应的任务,若否,则进入下一步骤;

  设置模块,所述设置模块用于根据所述综合性能指标需求和综合负载指标需求设置所述边缘节点的综合性能指标最大值、综合负载指标最大值、综合性能指标阈值、综合负载指标阈值和请求连接数阈值;

  第二判断模块,所述第二判断模块用于判断当前请求连接数是否超过所述请求连接数阈值,若否,则将所述当前请求连接数输入请求连接数预测模型,预测得到所述边缘节点的

下一时刻请求连接数,所述请求连接数预测模型为通过LSTM算法构建的模型;

计算模块,所述计算模块用于根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数;

第三判断模块,所述第三判断模块用于判断所述边缘节点请求连接数是否超过所述请求连接数阈值,若否,则判断所述综合性能指标是否超过所述综合性能指标最大值,若否则进入下一步骤;

第四判断模块,所述第四判断模块用于判断所述综合负载指标是否超过所述综合负载指标最大值,若否则进入下一步骤;

第五判断模块,所述第五判断模块用于判断所述综合性能指标需求是否超过所述综合性能指标阈值,若否则进入下一步骤;

第六判断模块,所述第六判断模块用于判断所述综合负载指标需求是否超过所述综合负载指标阈值,若否则进入下一步骤;

预测模块,所述预测模块用于将所述边缘节点的综合性能指标、综合负载指标和所述边缘节点请求连接数输入任务等待时间模型,输出预测结果,所述预测结果包括每个所述边缘节点的任务等待时间,所述任务等待时间模型为通过PSO-SVM算法构建的模型;

筛选模块,所述筛选模块用于对所述预测结果进行排序,筛选出所有所述边缘节点中任务等待时间最短的所述边缘节点,得到最佳边缘节点,通过所述最佳边缘节点部署所述任务请求对应的任务。

## 一种基于LSTM-SVM的物联网负载均衡任务调度方法及装置

### 技术领域

[0001] 本发明涉及电数字数据处理技术领域,尤其涉及一种基于LSTM-SVM的物联网负载均衡任务调度方法及装置。

### 背景技术

[0002] 静态负载均衡算法是指不考虑服务器的实时负载状态,仅利用一些统计数值或者数学函数以一定比例来分配请求,主要包含轮询调度算法、加权轮询调度算法、地址散列调度算法和一致性哈希算法。目前传统的静态负载均衡算法平均响应时间较长,集群吞吐率和请求响应成功率较低。

[0003] 为了解决上述问题,本发明研发出一种基于LSTM-SVM的物联网负载均衡任务调度方法来解决上述问题。

### 发明内容

[0004] 本发明提出一种基于LSTM-SVM的物联网负载均衡任务调度方法及装置,以解决现有静态负载均衡算法平均响应时间较长,集群吞吐率和请求响应成功率较低的问题。

[0005] 本发明通过以下技术方案来实现上述目的:

[0006] 本发明一方面提供了一种基于LSTM-SVM的物联网负载均衡任务调度方法,包括:

[0007] 接收任务请求;

[0008] 获取所述任务请求的综合性能指标需求和综合负载指标需求;

[0009] 获取边缘节点集群和边缘节点的综合性能指标、综合负载指标和当前请求连接数;

[0010] 判断所述任务请求是否指定所述边缘节点集群中的某个边缘节点作为指定边缘节点,若是,则通过所述指定边缘节点部署所述任务请求对应的任务,若否,则进入下一步骤;

[0011] 根据所述综合性能指标需求和综合负载指标需求设置所述边缘节点的综合性能指标最大值、综合负载指标最大值、综合性能指标阈值、综合负载指标阈值和请求连接数阈值;

[0012] 判断当前请求连接数是否超过所述请求连接数阈值,若否,则将所述当前请求连接数输入请求连接数预测模型,预测得到所述边缘节点的下一时刻请求连接数,所述请求连接数预测模型为通过LSTM算法构建的模型;

[0013] 根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数;

[0014] 判断所述边缘节点请求连接数是否超过所述请求连接数阈值,若否,则判断所述综合性能指标是否超过所述综合性能指标最大值,若否则进入下一步骤;

[0015] 判断所述综合负载指标是否超过所述综合负载指标最大值,若否则进入下一步骤;

- [0016] 判断所述综合性能指标需求是否超过所述综合性能指标阈值,若否则进入下一步骤;
- [0017] 判断所述综合负载指标需求是否超过所述综合负载指标阈值,若否则进入下一步骤;
- [0018] 将所述边缘节点的综合性能指标、综合负载指标和所述边缘节点请求连接数输入任务等待时间模型,输出预测结果,所述预测结果包括每个所述边缘节点的任务等待时间,所述任务等待时间模型为通过PSO-SVM算法构建的模型;
- [0019] 对所述预测结果进行排序,筛选出所有所述边缘节点中任务等待时间最短的所述边缘节点,得到最佳边缘节点,通过所述最佳边缘节点部署所述任务请求对应的任务。
- [0020] 进一步地,获取边缘节点集群,包括:
- [0021] 获取原始边缘节点集群;
- [0022] 在所述原始边缘节点集群中筛选出满足部署条件的所有原始边缘节点形成新边缘节点集群,所述部署条件为判断所述原始边缘节点是否拥有能够部署所述任务请求对应任务硬件和软件资源;
- [0023] 将所述新边缘节点集群作为所述边缘节点集群。
- [0024] 具体地,通过LSTM算法构建所述请求连接数预测模型,包括:
- [0025] 基于LSTM算法构建LSTM模型;
- [0026] 根据所述边缘节点的请求连接数历史数据和具体时刻构建输入矩阵;
- [0027] 对所述输入矩阵进行标准化处理,得到标准化数据;
- [0028] 将所述标准化数据划分为训练集和测试集;
- [0029] 根据所述训练集对所述LSTM模型进行训练,在训练过程中,将ReLU函数作为所述LSTM模型的激活函数,采用Hyperopt对所述LSTM模型进行模型参数调整,采用Dropout方法优化所述LSTM模型的层间过拟合程度,采用Adam方法对所述LSTM模型进行优化,得到所述请求连接数预测模型。
- [0030] 其中Hyperopt为用于超参数优化的python库,Dropout方法为丢弃法,Dropout方法为一种用于深度学习神经网络的正则化技术,旨在防止过拟合,Adam方法为一种基于梯度下降的优化算法,用于求解非线性方程组。
- [0031] 具体地,对所述输入矩阵进行标准化处理,得到标准化数据,包括:
- [0032] 计算所述输入矩阵的均值;
- [0033] 计算所述输入矩阵的方差;
- [0034] 计算所述输入矩阵与所述均值的差值;
- [0035] 计算所述差值与所述方差的比值,得到所述标准化数据。
- [0036] 具体地,通过PSO-SVM算法构建任务等待时间模型,包括:
- [0037] 基于SVM算法构建SVM模型;
- [0038] 采用PSO算法对SVM模型进行优化;
- [0039] 将所述边缘节点的性能指标历史数据、负载指标历史数据和请求连接数历史数据作为模型输入;
- [0040] 根据所述模型输入对优化后的所述SVM模型进行训练,得到所述任务等待时间模型。

[0041] 进一步地,采用PSO算法对SVM模型进行优化,包括:

[0042] 初始化粒子群,每个粒子表示一组SVM模型参数,所述粒子的位置表示SVM模型的惩罚系数,所述粒子的速度表示SVM模型的核函数;

[0043] 更新所述粒子的位置和速度,将预测平均误差值 $g[\gamma(t)]$ 作为确定所述粒子的最优位置的目标函数适应值,对所述SVM模型参数寻优,所述预测平均误差值的计算公式如下:

$$[0044] g[\gamma(t)] = \left( \frac{M_{\text{error}}}{M_{\text{total}}} * 100 \right),$$

[0045]  $M_{\text{error}}$ 为预测的时间数据的误差, $M_{\text{total}}$ 为预测数据的数量;

[0046] 所述粒子的位置更新公式为:

$$[0047] p_i(t+1) = \begin{cases} p_i(t), & g[\gamma(t+1)] \geq g[p_i(t)] \\ \gamma(t+1), & g[\gamma(t+1)] < g[p_i(t)] \end{cases}$$

[0048] 将 $P_i(t+1)$ 代入粒子群算法迭代公式:

$$[0049] v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)[p_{ij}(t) - x_{ij}(t)] + c_2r_{2j}(t)[p_{gj}(t) - x_{ij}(t)],$$

$$[0050] x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1),$$

[0051]  $P_i(t)$ 代表粒子*i*在当前时刻的位置, $P_i(t+1)$ 代表粒子*i*在下一时刻的位置, $g[\gamma(t+1)]$ 代表下一时刻的预测平均误差值, $g[P_i(t)]$ 表示粒子*i*当前的适应值, $\gamma(t+1)$ 表示下一时刻的待优化参数,其中w为惯性权重因子, $c_1, c_2$ 为加速因子; $r_{1j}(t), r_{2j}(t)$ 为在[0,1]之间相对独立变化的随机函数, $x_{ij}(t)$ 表示t时刻粒子*i*在第j维空间的位置, $x_{ij}(t+1)$ 表示t+1时刻粒子*i*在第j维空间的位置, $v_{ij}(t)$ 表示t时刻粒子*i*在第j维空间的速度,即每次迭代中粒子移动的距离, $v_{ij}(t+1)$ 表示t+1时刻粒子*i*在第j维空间的速度, $P_{gj}(t)$ 为单个粒子*i*在所优化的第j个参数的历史最优解, $P_{gj}(t)$ 表示所有粒子在所优化的第j参数中的历史最优解;

[0052] 根据所述粒子群算法迭代公式不断迭代调整更新粒子位置和速度,直到达到预设的停止条件为止,最终得到最优的SVM模型参数。

[0053] 进一步地,获取所述边缘节点的综合性能指标、综合负载指标,包括:

[0054] 获取所述边缘节点的CPU利用率、硬盘利用率、内存利用率和带宽利用率;

[0055] 根据所述边缘节点的CPU利用率、硬盘利用率、内存利用率和带宽利用率基于优序图法计算得到所述边缘节点的综合性能指标;

[0056] 获取所述边缘节点的CPU大小、硬盘大小、内存大小和带宽大小;

[0057] 根据所述边缘节点的CPU大小、硬盘大小、内存大小和带宽大小基于优序图法得到所述边缘节点的综合负载指标。

[0058] 进一步地,本发明一种基于LSTM-SVM的物联网负载均衡任务调度方法还包括对所述预测结果进行评估,对所述预测结果进行评估的步骤包括:

[0059] 获取在所述最佳边缘节点部署所述任务后,所述边缘节点集群中各个边缘节点的任务等待时间;

[0060] 根据所述边缘节点集群中各个边缘节点的任务等待时间计算所述边缘节点集群的平均任务等待时间;

[0061] 判断所述平均任务等待时间是否不超过预设阈值,所述预设阈值为所述边缘节点

集群负载均衡下的平均任务等待时间。

[0062] 具体地,根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数,包括:

[0063] 对所述当前请求连接数和所述下一时刻请求连接数进行重要性排序;

[0064] 根据排序后所述当前请求连接数和所述下一时刻请求连接数,基于优序图法计算所述当前请求连接数的第一权重和所述下一时刻请求连接数的第二权重;

[0065] 根据所述第一权重、第二权重、所述当前请求连接数和所述下一时刻请求连接数加权计算得到所述边缘节点请求连接数。

[0066] 本发明另一方面还提供了一种基于LSTM-SVM的物联网负载均衡任务调度装置,所述装置包括:

[0067] 接收模块,所述接收模块用于接收任务请求;

[0068] 第一获取模块,所述第一获取模块用于获取所述任务请求的综合性能指标需求和综合负载指标需求;

[0069] 第二获取模块,所述第二获取模块用于获取边缘节点集群和所述边缘节点的综合性能指标、综合负载指标和当前请求连接数;

[0070] 第一判断模块,所述第一判断模块用于判断所述任务请求是否指定所述边缘节点集群中的某个边缘节点作为指定边缘节点,若是,则通过所述指定边缘节点部署所述任务请求对应的任务,若否,则进入下一步骤;

[0071] 设置模块,所述设置模块用于根据所述综合性能指标需求和综合负载指标需求设置所述边缘节点的综合性能指标最大值、综合负载指标最大值、综合性能指标阈值、综合负载指标阈值和请求连接数阈值;

[0072] 第二判断模块,所述第二判断模块用于判断当前请求连接数是否超过所述请求连接数阈值,若否,则将所述当前请求连接数输入请求连接数预测模型,预测得到所述边缘节点的下一时刻请求连接数,所述请求连接数预测模型为通过LSTM算法构建的模型;

[0073] 计算模块,所述计算模块用于根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数;

[0074] 第三判断模块,所述第三判断模块用于判断所述边缘节点请求连接数是否超过所述请求连接数阈值,若否,则判断所述综合性能指标是否超过所述综合性能指标最大值,若否则进入下一步骤;

[0075] 第四判断模块,所述第四判断模块用于判断所述综合负载指标是否超过所述综合负载指标最大值,若否则进入下一步骤;

[0076] 第五判断模块,所述第五判断模块用于判断所述综合性能指标需求是否超过所述综合性能指标阈值,若否则进入下一步骤;

[0077] 第六判断模块,所述第六判断模块用于判断所述综合负载指标需求是否超过所述综合负载指标阈值,若否则进入下一步骤;

[0078] 预测模块,所述预测模块用于将所述边缘节点的综合性能指标、综合负载指标和所述边缘节点请求连接数输入任务等待时间模型,输出预测结果,所述预测结果包括每个所述边缘节点的任务等待时间,所述任务等待时间模型为通过PSO-SVM算法构建的模型;

[0079] 筛选模块,所述筛选模块用于对所述预测结果进行排序,筛选出所有所述边缘节

点中任务等待时间最短的所述边缘节点,得到最佳边缘节点,通过所述最佳边缘节点部署所述任务请求对应的任务。

[0080] 本发明的有益效果在于:

[0081] 本发明提出的基于LSTM-SVM的物联网负载均衡任务调度方法综合考虑了边缘节点的性能指标,实时负载信息以及服务器的流量突发性,解决了高并发请求给边缘节点节点带来的压力,为用户提供了更好的服务体验相比于传统的负载均衡方案,本文提出的负载均衡方案不仅提高了边缘节点集群吞吐率和请求响应成功率,还大幅度缩短了并发请求的平均等待时间。

## 附图说明

[0082] 图1是本发明实施例中的负载均衡方法流程图。

## 具体实施方式

[0083] 为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述。显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。通常在此处附图中描述和示出的本发明实施例的组件可以以各种不同的配置来布置和设计。

[0084] 因此,以下对在附图中提供的本发明的实施例的详细描述并非旨在限制要求保护的本发明的范围,而是仅仅表示本发明的选定实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0085] 应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。

[0086] 在本发明的描述中,需要理解的是,术语“上”、“下”、“内”、“外”、“左”、“右”等指示的方位或位置关系为基于附图所示的方位或位置关系,或者是该发明产品使用时惯常摆放的方位或位置关系,或者是本领域技术人员惯常理解的方位或位置关系,仅是为了便于描述本发明和简化描述,而不是指示或暗示所指的设备或元件必须具有特定的方位、以特定的方位构造和操作,因此不能理解为对本发明的限制。

[0087] 此外,术语“第一”、“第二”等仅用于区分描述,而不能理解为指示或暗示相对重要性。

[0088] 在本发明的描述中,还需要说明的是,除非另有明确的规定和限定,“设置”、“连接”等术语应做广义理解,例如,“连接”可以是固定连接,也可以是可拆卸连接,或一体地连接;可以是机械连接,也可以是电连接;可以是直接连接,也可以通过中间媒介间接连接,可以是两个元件内部的连通。对于本领域的普通技术人员而言,可以根据具体情况理解上述术语在本发明中的具体含义。

[0089] 下面结合附图,对本发明的具体实施方式进行详细说明。

[0090] 如图1所示,一种基于LSTM-SVM的物联网负载均衡任务调度方法,包括:

[0091] 接收任务请求;

[0092] 获取所述任务请求的综合性能指标需求和综合负载指标需求;

- [0093] 获取边缘节点集群和所述边缘节点的综合性能指标、综合负载指标和当前请求连接数；
- [0094] 判断所述任务请求是否指定所述边缘节点集群中的某个边缘节点作为指定边缘节点，若是，则通过所述指定边缘节点部署所述任务请求对应的任务，若否，则进入下一步骤；
- [0095] 根据所述综合性能指标需求和综合负载指标需求设置所述边缘节点的综合性能指标最大值、综合负载指标最大值、综合性能指标阈值、综合负载指标阈值和请求连接数阈值；
- [0096] 判断当前请求连接数是否超过所述请求连接数阈值，若否，则将所述当前请求连接数输入请求连接数预测模型，预测得到所述边缘节点的下一时刻请求连接数，所述请求连接数预测模型为通过LSTM算法构建的模型；
- [0097] 根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数；
- [0098] 判断所述边缘节点请求连接数是否超过所述请求连接数阈值，若否，则判断所述综合性能指标是否超过所述综合性能指标最大值，若否则进入下一步骤；
- [0099] 判断所述综合负载指标是否超过所述综合负载指标最大值，若否则进入下一步骤；
- [0100] 判断所述综合性能指标需求是否超过所述综合性能指标阈值，若否则进入下一步骤；
- [0101] 判断所述综合负载指标需求是否超过所述综合负载指标阈值，若否则进入下一步骤；
- [0102] 将所述边缘节点的综合性能指标、综合负载指标和所述边缘节点请求连接数输入任务等待时间模型，输出预测结果，所述预测结果包括每个所述边缘节点的任务等待时间，所述任务等待时间模型为通过PSO-SVM算法构建的模型；
- [0103] 对所述预测结果进行排序，筛选出所有所述边缘节点中任务等待时间最短的所述边缘节点，得到最佳边缘节点，通过所述最佳边缘节点部署所述任务请求对应的任务。
- [0104] 当前请求连接数：这是直接影响到边缘节点当前负载的直接因素。如果当前请求连接数较高，意味着边缘节点需要更多的资源来处理这些请求，否则可能会导致服务延迟或中断。下一时刻请求连接数的预测：通过对未来请求连接数的预测，边缘节点可以提前准备和优化资源配置。例如，如果预测到下一时刻会有大量的请求涌入，边缘节点可以提前增加资源分配，确保服务能够平稳处理这些请求，避免因资源不足而导致的服务降级。综上所述，边缘节点的请求连接数不仅受到当前请求的影响，还需要对下一时刻的请求进行预测。
- [0105] 本发明提供的上述判断流程顺序的好处在于：
- [0106] 1. 提高处理效率：
- [0107] 直接对每个元素与阈值进行比较可能会非常耗时。如果先确定最大值，那么可以迅速判断整个数据集是否有可能超过某个特定的阈值。如果最大值都小于阈值，那么就没有必要再对剩余的元素进行逐一比较，从而节省了大量的计算资源。
- [0108] 2. 逻辑清晰性：

[0109] 先确定最大值可以帮助我们更好地理解数据的分布情况。了解数据的最大值可以为我们提供一个直观的参考点,从而更容易地设定合理的阈值。此外,这种顺序也符合人类处理问题的直觉,即先了解整体情况,再做出具体判断。

[0110] 3减少误判率 :

[0111] 直接对每个元素与阈值进行比较可能会导致较高的误判率。如果数据中存在大量的异常值,那么这些值可能会错误地触发阈值条件。通过先确定最大值,我们可以更准确地识别出这些异常值,并在必要时对它们进行特殊处理,从而降低误判率。

[0112] 4.支持并行处理 :

[0113] 先确定最大值可以作为一个预处理步骤,为后续的并行处理提供有用的信息。每个节点可以先计算本地数据的最大值,然后将这些最大值汇总到中心节点以计算全局最大值。这样,后续的阈值判断就可以基于全局最大值进行,从而支持更高效的并行处理。

[0114] 如图1所示,在一些实施例中,获取边缘节点集群,包括:

[0115] 获取原始边缘节点集群;

[0116] 在所述原始边缘节点集群中筛选出满足部署条件的所有原始边缘节点形成新边缘节点集群,所述部署条件为判断所述原始边缘节点是否拥有能够部署所述任务请求对应任务硬件和软件资源;

[0117] 将所述新边缘节点集群作为所述边缘节点集群。

[0118] 先剔除不满足部署条件的边缘节点,由此能够减少本方法后续的数据处理量。

[0119] 在一些实施例中通过LSTM算法构建所述请求连接数预测模型,包括:

[0120] 基于LSTM算法构建LSTM模型;

[0121] 根据所述边缘节点的请求连接数历史数据和具体时刻构建输入矩阵;

[0122] 对所述输入矩阵进行标准化处理,得到标准化数据;

[0123] 将所述标准化数据划分为训练集和测试集;

[0124] 根据所述训练集对所述LSTM模型进行训练,在训练过程中,将所述ReLU函数作为所述LSTM模型的激活函数,采用Hyperopt对所述LSTM模型进行模型参数调整,采用Dropout方法优化所述LSTM模型的层间过拟合程度,采用Adam方法对所述LSTM模型进行优化,得到所述请求连接数预测模型。

[0125] 在一些实施例中,对所述输入矩阵进行标准化处理,得到标准化数据,包括:

[0126] 计算所述输入矩阵的均值;

[0127] 计算所述输入矩阵的方差;

[0128] 计算所述输入矩阵与所述均值的差值;

[0129] 计算所述差值与所述方差的比值,得到所述标准化数据。

[0130] 在一些实施例中,通过PSO-SVM算法构建任务等待时间模型,包括:

[0131] 基于SVM算法构建SVM模型;

[0132] 采用PSO算法对SVM模型进行优化;

[0133] 将所述边缘节点的性能指标历史数据、负载指标历史数据和请求连接数历史数据作为构建模型输入;

[0134] 根据所述模型输入对所述PSO-SVM模型进行训练,得到所述任务等待时间模型。

[0135] 使用SVM模型预测节点任务等待时间,其中创建SVM模型是基于高维数特征空间知

识构建将输入变量 $x$ 进行非线性变换,通过非线性映射把输入空间 $R^n$ 中的数据映射到一个高维特征空间 $F$ 中,非线性映射的基本结构为:

[0136]  $\varphi(x): R^n \rightarrow F,$

[0137]  $x \rightarrow \varphi(x),$

[0138] 则目标函数 $y=f(x)=w\varphi(x)+b$ 式中 $y$ 是最终预测值, $w$ 和 $b$ 是函数的参数向量; $\varphi(x)$ 表示非线性变换函数,通过求解下列的最优问题得出预测值。

[0139] 设训练数据集为 $(x_i, y_i)$ ,其中,

$i = 1, 2, \dots, N$ ,  $y_i \in (0, 20)$ ,  $x_i \in R^N$ ,  $\gamma(t) = \{\sigma(t), C(t)\}$ 其中 $N$ 为样本数量, $x_i$ 是输入量, $x_i = (x_{1i}, x_{2i}, x_{3i}, \dots)^T$ ,其中 $x_{1i}, x_{2i}, x_{3i}, \dots$ 为输入量的要素。

[0140] 这里选取边缘节点的性能指标 $P(N_i)$ 节点负载 $L(N_i)$ 和边缘节点请求连接数 $Rank(N_i)$ 作为SVM模型的输入量,即用 $x_i = (P(N_i), L(N_i), Rank(N_i))^T$ 表示。 $y_i, y_j$ 均为类标记, $y_i, y_j$ 的单位均为毫秒(ms), $\alpha_i$ 和 $\alpha_j$ 均为拉格朗日乘子, $\sigma(t)$ 是核函数, $C(t)$ 是惩罚系数。 $\gamma(t)$ 是待优化参数在迭代过程中的参数优化过程数值,选用径向量核函数,并且将 $\gamma(t)$ 代入,构造并求解如下的最优问题:

[0141]  $\min \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \exp\left\{-\frac{|x-x_j|^2}{\sigma(t)^2}\right\} - \sum_{i=1}^N \alpha_i,$

[0142] s.t.  $\sum_{i=1}^N y_i \alpha_i = 0,$

[0143]  $0 <= \alpha_i <= C(t), i = 1, \dots, N,$

[0144] 求得一组Lagrange乘子系数解 $\alpha^* = (\alpha_1^*, \dots, \alpha_N^*)^T$ 。选取 $\alpha^*$ 中的一个正分量 $0 < \alpha_i^* < C(t)$ ,并以此计算阈值 $b^*$ :

[0145]  $b^* = y_i - \sum_{i=1}^N y_i \alpha_i^* \exp\left\{-\frac{|x-x_i|^2}{\sigma(t)^2}\right\},$

[0146] 由此可以构建决策函数即回归函数(目标函数)

[0147]  $f(x) = \text{sgn}\left(\sum_{i=1}^N \alpha_i^* y_i \exp\left\{-\frac{|x-x_i|^2}{\sigma(t)^2}\right\} + b^*\right),$

[0148] 根据这个目标函数即可求解任务等待时间。

[0149] 在一些实施例中,采用PSO算法对SVM模型进行优化,包括:

[0150] 初始化粒子群,每个粒子表示一组SVM模型参数,所述粒子的位置表示SVM模型的惩罚系数,所述粒子的速度表示SVM模型的核函数;

[0151] 更新所述粒子的位置和速度,将预测平均误差值 $g[\gamma(t)]$ 作为确定所述粒子的最优位置的目标函数适应值,对所述SVM模型参数寻优,所述预测平均误差值的计算公式如下:

[0152]  $g[\gamma(t)] = \left(\frac{M_{\text{error}}}{M_{\text{total}}} * 100\right),$

[0153]  $M_{\text{error}}$ 为预测的时间数据的误差, $M_{\text{total}}$ 为预测数据的数量;

[0154] 所述粒子的位置更新公式为:

$$[0155] p_i(t+1) = \begin{cases} p_i(t), g[\gamma(t+1)] \geq g[p_i(t)] \\ \gamma(t+1), g[\gamma(t+1)] < g[p_i(t)], \end{cases}$$

[0156] 将 $P_i(t+1)$ 代入粒子群算法迭代公式:

$$[0157] v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)[p_{ij}(t) - x_{ij}(t)] + c_2r_{2j}(t)[p_{gj}(t) - x_{ij}(t)],$$

$$[0158] x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1),$$

[0159]  $P_i(t)$ 代表粒子*i*在当前时刻的位置, $P_i(t+1)$ 代表粒子*i*在下一时刻的位置, $g[\gamma(t+1)]$ 代表下一时刻的预测平均误差值, $g[P_i(t)]$ 表示粒子*i*当前的适应值, $\gamma(t+1)$ 表示下一时刻的待优化参数,其中w为惯性权重因子, $c_1, c_2$ 为加速因子; $r_{1j}(t), r_{2j}(t)$ 为在[0,1]之间相对独立变化的随机函数, $x_{ij}(t)$ 表示t时刻粒子*i*在第j维空间的位置, $x_{ij}(t+1)$ 表示t+1时刻粒子*i*在第j维空间的位置, $v_{ij}(t)$ 表示t时刻粒子*i*在第j维空间的速度,即每次迭代中粒子移动的距离, $v_{ij}(t+1)$ 表示t+1时刻粒子*i*在第j维空间的速度, $P_{gj}(t)$ 为单个粒子*i*在所优化的第j个参数的历史最优解, $P_{gj}(t)$ 表示所有粒子在所优化的第j参数中的历史最优解;

[0160] 根据所述粒子群算法迭代公式不断迭代调整更新粒子位置和速度,直到达到预设的停止条件为止,最终得到最优的SVM模型参数。

[0161] 为了评估本发明,以下通过以下四个公式作为评估标准:

$$[0162] MAE = \sum_{i=1}^N \frac{|U_i - V_i|}{N},$$

$$[0163] RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (U_i - V_i)^2},$$

$$[0164] NSE = 1 - \frac{\sum_{i=1}^N (U_i - V_i)^2}{\sum_{i=1}^N (U_i - \bar{U})^2},$$

$$[0165] R^2 = 1 - \frac{\sum_{i=1}^N (U_i - \hat{V}_i)^2}{\sum_{i=1}^N (U_i - \bar{U})^2},$$

[0166] 其中 $U_i$ 为实际值, $\bar{U}$ 是实际值均值, $V_i$ 为模型预测值, $\hat{V}_i$ 为统计模型的变量模拟值;N为样本量。模型性能评级标准如表1。

[0167] 表1

性能评级	$R^2$	NSE
非常好	$0.75 < R^2 \leq 1.00$	$0.75 < NSE \leq 1.00$
良好	$0.65 < R^2 \leq 0.75$	$0.65 < NSE \leq 0.75$
符合要求	$0.50 < R^2 \leq 0.65$	$0.50 < NSE \leq 0.65$
不符合要求	$R^2 \leq 0.50$	$NSE \leq 0.50$

[0169] 平均绝对误差(MAE)和均方误差(RMSE)越接近0,模型的性能越好;Nash-Sutcliffe模型的效率系数(NSE)和决定系数( $R^2$ )越接近1,模型的性能越好。

[0170] 在一些实施例中,获取所述边缘节点的综合性能指标、综合负载指标,包括:

- [0171] 获取所述边缘节点的CPU利用率、硬盘利用率、内存利用率和带宽利用率；  
[0172] 根据所述边缘节点的CPU利用率、硬盘利用率、内存利用率和带宽利用率基于优序图法计算得到所述边缘节点的综合性能指标；  
[0173] 获取所述边缘节点的CPU大小、硬盘大小、内存大小和带宽大小；  
[0174] 根据所述边缘节点的CPU大小、硬盘大小、内存大小和带宽大小基于优序图法得到所述边缘节点的综合负载指标。  
[0175] 针对不同的任务类型的影响参数进行不同优先级排序，CPU密集型任务则CPU优先级最高，带宽密集型任务则带宽优先级最高。  
[0176] 在一些实施例中，还包括对所述预测结果进行评估，对所述预测结果进行评估的步骤包括：  
[0177] 获取在所述最佳边缘节点部署所述任务后，所述边缘节点集群中各个边缘节点的任务等待时间；  
[0178] 根据所述边缘节点集群中各个边缘节点的任务等待时间计算所述边缘节点集群的平均任务等待时间；  
[0179] 判断所述平均任务等待时间是否不超过预设阈值，所述预设阈值为所述边缘节点集群负载均衡下的平均任务等待时间。

[0180] 假设总共边缘集群共有N个边缘节点，其中边缘节点的的任务等待时间为 $T(N_i)$ ，  
则所有边缘节点的平均任务等待时间为 $T_{average} = \frac{\sum_{i=1}^N T(N_i)}{N}$ 。任务等待时间越小，分

配任务请求的可能性越大。任务等待时间与边缘节点的性能成正比，边缘节点的性能指标 $P(N_i)$ 越高，可以处理的请求越多。但是随着边缘节点的长时间运行，服务器的剩余性能将越来越少，这一指标将有所局限，因此我们引入了边缘节点的实时负载 $L(N_i)$ ，边缘节点的负载与任务等待时间成反比，负载越高，边缘节点可以处理的请求越少。此外，不同的边缘节点具有不同的性能，处理相同数量的请求对于高性能的边缘节点来说，可能只是占用了一少部分资源，但是对于低性能的边缘节点来说，可能已经占据了大部分资源，这样也是负载不均衡，因此发本明引入了请求连接数这一指标，该指标与任务等待时间成反比。综上所述，我们用边缘节点集群的平均任务等待时间评估边缘节点集群的负载均衡。

[0181] 边缘节点集群的负载均衡主要关注如何合理分配任务或数据流量到多个计算资源上，以达到提高系统性能、增加系统可靠性、提供可扩展性的目的。平均任务等待时间作为评估负载均衡效果的一个指标，可以反映系统对任务分配的效率和公平性。如果平均任务等待时间较短，说明任务能够被快速且均匀地分配到各个节点上，从而实现了较好的负载均衡。反之，如果平均任务等待时间较长，则可能意味着某些节点过载，而其他节点空闲，这表明负载均衡存在问题。

[0182] 在一些实施例中，根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数，包括：

- [0183] 对所述当前请求连接数和所述下一时刻请求连接数进行重要性排序；  
[0184] 根据排序后所述当前请求连接数和所述下一时刻请求连接数，基于优序图法计算所述当前请求连接数的第一权重和所述下一时刻请求连接数的第二权重；  
[0185] 根据所述第一权重、第二权重、所述当前请求连接数和所述下一时刻请求连接数

加权计算得到所述边缘节点请求连接数。

[0186] SVM模型特征向量选择,本发明选取边缘节点性能指标P(N<sub>i</sub>)、负载指标L(N<sub>i</sub>)和边缘节点请求连接数Rank(N<sub>i</sub>)这3种边缘节点特征作为输入特征向量。

[0187] 其中  $P(N_i) = a1 \frac{Cs_i}{\sum_{i=1}^N Cs_i} + a2 \frac{Ds_i}{\sum_{i=1}^N Ds_i} + a3 \frac{Ms_i}{\sum_{i=1}^N Ms_i} + a4 \frac{Bs_i}{\sum_{i=1}^N Bs_i}$ ,

[0188] a1+a2+a3+a4=1,其中a1为Cs<sub>i</sub>的权重,a2为Ds<sub>i</sub>的权重,a3为Ms<sub>i</sub>的权重,a4表示Bs<sub>i</sub>的权重;

[0189]  $L(N_i) = b1Cu + b2Du + b3Mu + b4Bu$ ,

[0190] b1+b2+b3+b4=1,其中b1为Cu的权重,b2为Du的权重,b3表示Mu的权重,b4表示Bu的权重;

[0191] 其中,Cu表示CPU利用率,Du表示硬盘利用率,Mu表示内存利用率,Bu表示带宽利用率,Cs<sub>i</sub>表示CPU大小,Ds<sub>i</sub>表示硬盘大小,Ms<sub>i</sub>表示内存大小,Bs<sub>i</sub>表示带宽大小。

[0192] 边缘节点请求连接数不仅与下一时刻的连接数q<sub>t+1</sub>(N<sub>i</sub>)有关,还与当前时刻的连接数q<sub>t</sub>(N<sub>i</sub>)有很大关系,因此边缘节点请求连接数Rank(N<sub>i</sub>)为:

[0193]  $Rank(N_i) = c1q_{t+1}(N_i) + c2q_t(N_i)$ ,

[0194] c1+c2=1;

[0195] 其中权重设定由优序图法决定,AHP层次分析法和优序图法都为主观赋权法,这两种方法利用数据的大小信息进行权重计算;优序图法的原理是用矩阵图示的办法两两比较分析各因素对目标的重要程度。由于层次分析法在实际应用时经常遇到判断矩阵不能通过一致性检验的问题,相比之下,优序图法更易理解、操作简便、结果可信度也较高。优序图法流程如下:

[0196] 对系统指标X1、X2、……Xn进行重要性排序(以节点性能指标为例)其中针对不同的类型的任务有不同的排序其中CPU密集型任务CPU>带宽>磁盘>内存,磁盘I0密集型任务磁盘>内存>CPU>带宽,网络I0密集型任务带宽>CPU>磁盘>内存。下列以CPU密集型任务举例。

[0197] 画棋盘图,并进行两两对比,专家对指标进行两两对比,若指标Xi比指标Xj重要,则Xi得1分;若同等重要,则Xi得0.5分;若指标Xj比指标Xi重要,则Xi得0分,对比结果如表2:

[0198] 表2

[0199]

	CPU	带宽	磁盘	内存	指标得分
CPU	0.5	1	1	1	3.5
带宽	0	0.5	1	1	2.5
磁盘	0	0	0.5	1	1.5
内存	0	0	0	0.5	0.5

[0200] 指标权重确定:

[0201] 则CPU权重为  $a1 = b1 = \frac{7}{16}$ , 带宽权重为  $a4 = b4 = \frac{5}{16}$ , 磁盘权重为

$a2 = b2 = \frac{3}{16}$ , 内存权重为  $a3 = b3 = \frac{1}{16}$ 。

[0202] 综上面向CPU密集型任务时,  $P(N_i) = \frac{7}{16} \frac{Cs_i}{\sum_{i=1}^N Cs_i} + \frac{3}{16} \frac{Ds_i}{\sum_{i=1}^N Ds_i} + \frac{1}{16} \frac{Ms_i}{\sum_{i=1}^N Ms_i} + \frac{5}{16} \frac{Bs_i}{\sum_{i=1}^N Bs_i}$ ,

$$L(N_i) = \frac{7}{16} Cu + \frac{3}{16} Du + \frac{1}{16} Mu + \frac{5}{16} Bu \cdot \text{Rank}(N_i)$$

$$= 1/2q_{t+1}(N_i) + 1/2q_t(N_i)$$

[0203] 其他边缘节点的请求连接数和负载指标权重计算原理同上。

[0204] 采用优序图法来计算边缘节点性能指标和负载的指标的权重, 针对不同任务类型有不同的指标权重, 极大地提高了数据处理速度和精度。

[0205] 本发明实施例还提供了一种基于LSTM-SVM的物联网负载均衡任务调度装置, 其特征在于, 所述装置包括:

[0206] 接收模块, 所述接收模块用于接收任务请求;

[0207] 第一获取模块, 所述第一获取模块用于获取所述任务请求的综合性能指标需求和综合负载指标需求;

[0208] 第二获取模块, 所述第二获取模块用于获取边缘节点集群和所述边缘节点的综合性能指标、综合负载指标和当前请求连接数;

[0209] 第一判断模块, 所述第一判断模块用于判断所述任务请求是否指定所述边缘节点集群中的某个边缘节点作为指定边缘节点, 若是, 则通过所述指定边缘节点部署所述任务请求对应的任务, 否则, 则进入下一步骤;

[0210] 设置模块, 所述设置模块用于根据所述综合性能指标需求和综合负载指标需求设置所述边缘节点的综合性能指标最大值、综合负载指标最大值、综合性能指标阈值、综合负载指标阈值和请求连接数阈值;

[0211] 第二判断模块, 所述第二判断模块用于判断当前请求连接数是否超过所述请求连接数阈值, 若否, 则将所述当前请求连接数输入请求连接数预测模型, 预测得到所述边缘节点的下一时刻请求连接数, 所述请求连接数预测模型为通过LSTM算法构建的模型;

[0212] 计算模块, 所述计算模块用于根据所述当前请求连接数和所述下一时刻请求连接数基于加权算法计算出边缘节点请求连接数;

[0213] 第三判断模块, 所述第三判断模块用于判断所述边缘节点请求连接数是否超过所述请求连接数阈值, 若否, 则判断所述综合性能指标是否超过所述综合性能指标最大值, 若否则进入下一步骤;

[0214] 第四判断模块, 所述第四判断模块用于判断所述综合负载指标是否超过所述综合负载指标最大值, 若否则进入下一步骤;

[0215] 第五判断模块, 所述第五判断模块用于判断所述综合性能指标需求是否超过所述综合性能指标阈值, 若否则进入下一步骤;

[0216] 第六判断模块, 所述第六判断模块用于判断所述综合负载指标需求是否超过所述综合负载指标阈值, 若否则进入下一步骤;

[0217] 预测模块, 所述预测模块用于将所述边缘节点的综合性能指标、综合负载指标和

所述边缘节点请求连接数输入任务等待时间模型,输出预测结果,所述预测结果包括每个所述边缘节点的任务等待时间,所述任务等待时间模型为通过PSO-SVM算法构建的模型;

[0218] 筛选模块,所述筛选模块用于对所述预测结果进行排序,筛选出所有所述边缘节点中任务等待时间最短的所述边缘节点,得到最佳边缘节点,通过所述最佳边缘节点部署所述任务请求对应的任务。

[0219] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明技术原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

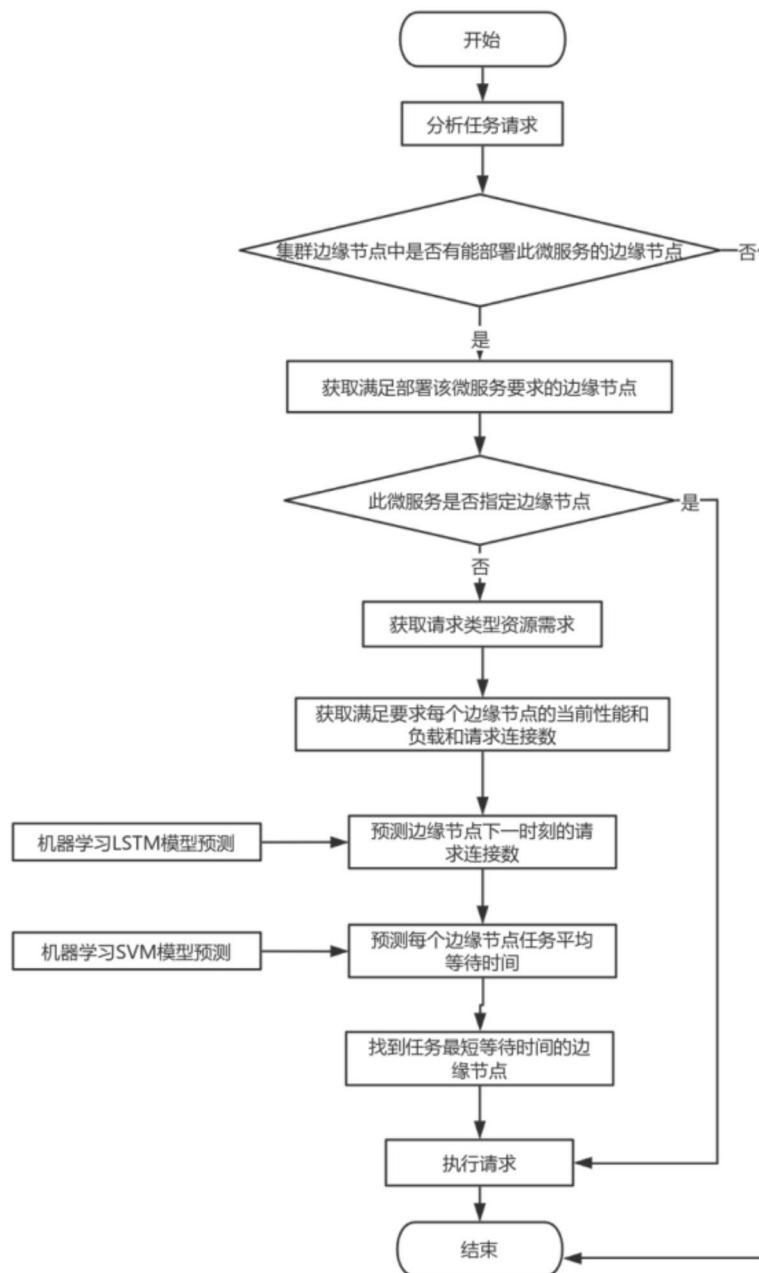


图 1