

高性能并行计算中节点负载均衡分配算法

田彦明¹ 侯永生¹ 高 超¹
TIAN Yanming HOU Yongsheng GAO Chao

摘 要

由于计算节点的处理能力、网络延迟、数据分布不均等因素，各节点的负载往往存在差异，这种负载不均衡不仅会导致系统资源的浪费，还可能引发某些节点过载，从而影响整个系统的性能和稳定性。因此，现提出高性能并行计算中节点负载均衡分配算法。首先，构建节点负载模型，反映系统资源的分配与利用情况，以达成更高效的负载平衡目标；其次，规划高性能并行计算熵，追求最大化熵增，实现最小化所有任务的总体执行时间，进而实现节点负载的均衡分配。实验结果表明：高性能并行计算中节点负载均衡分配算法能够在固定测试时间内最大程度上对节点负载进行均衡分配，而另外两种方法的分配数量较少，因此可以证明所提方法的有效性。

关键词

均衡分配；并行计算；负载均衡；节点；高性能

doi: 10.3969/j.issn.1672-9528.2024.11.009

0 引言

随着信息技术的飞速发展，高性能并行计算已成为处理大规模数据和复杂任务的关键技术。在现代计算机系统中，并行计算通过利用多个处理器或核心同时执行任务，显著提高了计算能力和效率。高性能并行计算系统通常由多个计算节点组成，每个节点都具备独立的处理能力和内存资源。负载均衡分配算法旨在优化任务分配策略，减少计算节点之间的负载差异，提高资源利用率，从而缩短计算时间并提升系统吞吐量。文献[1]采用 AGCH 算法对无线激光通信网络中的节点实施分组与成簇策略，进而识别出簇头节点，以此为基础构建了一个资源调度模型。随后引入自适应遗传算法求解此模型，旨在显著提升无线激光通信网络的负载均衡能力，实现网络的稳定高效运行。文献[2]选取了 CPU 效率、内存利用率、磁盘 I/O 速度及网络带宽等关键性能指标作为节点性能的评价基准，并依据节点性能使用率相对于集群平均负载使用率的变化动态调整算法权重，以灵活应对网络流量激增，确保服务器集群的整体性能与稳定性。而本文方法结合人工智能、机器学习等技术，实现更加智能的任务分配和负载均衡，根据系统状态的实时变化，动态调整负载均衡策略，以适应不同的应用场景和需求。在任务分配过程中，考虑更多的因素，以实现更精细化的负载均衡。

1 高性能并行计算中节点负载均衡分配算法的设计

1.1 构建节点负载模型

在计算机领域中，负载均衡技术是基于网络架构的一

种策略，其核心在于智能化重新分配已分配给各节点的处理任务，旨在实现各节点间负载的均衡化。这一过程有效规避了资源闲置与过载等待的并存现象，显著提升了系统资源的使用效率^[3]。同时，通过减少任务处理的平均等待时间，负载均衡分配机制进一步增强了系统的整体响应能力和性能表现。因此，在高性能并行计算中节点负载均衡分配算法的设计中，首先构建节点负载模型。

为优化高性能并行计算环境下的节点负载分配均衡性，引入了特征相关性分析技术^[4]。该技术基于节点信道的扩频序列特性，同时融合了多径损失的有效补偿机制与信息码元的智能重组策略，构建了一个创新的网络信道负载均衡模型。该模型的具体表达形式旨在精准反映系统资源的分配与利用情况，以达成更高效的负载平衡目标，则该模型的表达式为：

$$X_i = \frac{a_c}{\sum_i \left(x_i - \frac{w - a_c}{a_c} \right)^2} \quad (1)$$

式中： w 代表网络中的节点数， x_i 代表簇首节点 i 的成员节点数， a_c 代表簇首节点的数量。

运用抽头式传感器技术来达成网络节点间信道的均衡配置，该方法涉及自适应地调节均衡器的抽头系数。通过深入分析无线传感网络的信道承载特性^[5]，能够获取并计算出用于负载均衡调度的关键参数，其表达式为：

$$e_i = \frac{\sum_{k \in w} \alpha k_i \times t_k}{a_i} \quad (2)$$

1. 沈阳发动机研究所 辽宁沈阳 110000

式中: t_k 代表节点 k 所产生的总流量负载, αk_i 代表节点 k 的负载流量中需要通过网关 i 进行发送的部分。

运用自相关性校正与临界波束削弱技术, 实现网络信道内节点分配的优化配置。该网络由总数为 $w=2r$ 的阵列元素构建而成。在此基础上, 定义了无线传感网络信道的输出可靠性评估函数, 其表达式为:

$$f = X_i \cdot \frac{y_i(u) - x_p(u)}{h_i^2} \quad (3)$$

式中: $x_p(u)$ 代表信道估计误差, $y_i(u)$ 表示网络信道的偏移特征量。

在恒参信道 (即信道传输特性在传输过程中不发生变化的信道) 的稳态分配模型框架下^[6], 网络信道传输的多径分量为:

$$g^{ik} = \frac{f y_i(u)}{[x_p(u) h_i - \delta]} \quad (4)$$

式中: δ 代表引力常数。

通过整合多源拓扑节点的信息融合策略, 构建了网络信道的负载预测模型^[7]。此模型基于多元数据的综合处理, 旨在精确评估信道负载状况。在模型构建过程中, 推导出了信道负载估计的误差计算公式为:

$$\partial_i = \frac{2\pi h_i}{g^{ik}} \left(\sqrt{1 + \frac{1}{h_i^2}} - jp \right) \quad (5)$$

式中: p 代表元素间隔, j 代表网络信道分配节点的间隔距离。

基于网络信道理论上的速率衰减考量, 通过综合运用信号调制技术与扩频策略的分析^[8], 构建了一个反映网络信道承载能力的负载模型, 具体结构如图 1 所示。

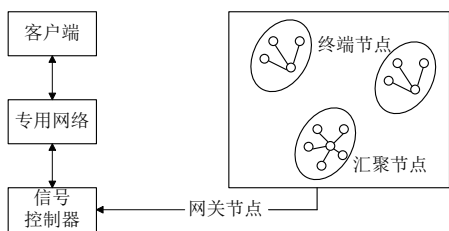


图 1 网络信道负载模型

1.2 规划高性能并行计算熵

高性能并行计算是一种利用多个处理器或核心同时执行任务以提高计算能力的技术。而并行计算熵衍生于 Shannon 信息熵理论, 其显著特性包括对称性、非负属性及可扩展性^[9]。在任务迁移策略中, 每一步均追求最大化熵增, 旨在最小化所有任务的总体执行时间。随着并行计算熵的增大, 这一过程有效减轻了负载最大的物理节点上的计算负担, 同时促进了其他节点间负载的均衡分布, 进而可知, 该优化策略导致

所有任务的总执行时间显著缩短。因此, 负载均衡分配的核心策略在于积极提升高性能并行计算熵, 以此作为减少总体任务执行时间的关键途径。

针对并行计算熵的概念, 其原始构想聚焦于同构集群与网格环境的应用。为拓宽其适用范围, 本文将该理论框架引入更为复杂的高性能异构集群之中^[10]。在此扩展下, 节点 i 的负载 t_i 的表达式为:

$$t_i = \sum_{j=1}^n \mu_{ij} d_j \quad (6)$$

式中: n 代表节点 i 上运行的所有任务数量, μ_{ij} 代表类型为 j 的节点负载模型所占的比重^[11-12], d_j 代表类型为 j 的节点负载模型的负载的大小。

则节点 i 的相对负载 T_i 为某一节点负载与所有节点的负载之比, 其计算公式为:

$$T_i = \frac{t_i}{\sum_{i=1}^w t_i} \quad (7)$$

鉴于不同节点的处理效能, 具体体现在其 CPU 性能、内存容量及网络带宽的差异性上^[13], 且这些参数的提升直接促进了节点处理能力的优化。因此, 可以设定一个假设, 即将节点的综合处理能力抽象表达为这三个关键参数的函数, 其表达公式如下:

$$S_i = \{S_{CPU_i}, S_{use_i}, S_{band_i}\} \quad (8)$$

式中: S_{CPU_i} 、 S_{use_i} 、 S_{band_i} 分别代表节点 CPU、内存和带宽。

对上述公式 (8) 进行量化, 则表达式为:

$$S_{total} = z_1 S_{CPU_i} + z_2 S_{use_i} + z_3 S_{band_i} \quad (9)$$

式中: z_1 、 z_2 、 z_3 均代表常数。

则节点的相对负载的表达式为:

$$T'_i = \frac{\frac{t_i}{S_i}}{\sum_{i=1}^w \frac{t_i}{S_i}} \quad (10)$$

在高性能并行计算中, 若存在 m 个节点构成的系统, 对于任意给定时间点 t , 则节点 i 的相对负载状况可使用来表示。基于这一情境, 高性能并行计算熵的规划结果为:

$$B(t) = \sum_{i=1}^w T'_i(t) \ln \left(\frac{1}{T'_i(t)} \right) \quad (11)$$

通过上述步骤, 即可完成对高性能并行计算熵的规划设计。

1.3 实现节点负载的均衡分配

负载均衡的目标是将工作负载 (如请求、任务或数据)

在多个计算节点或服务器之间有效分配^[14-15]，以优化资源利用率、减少响应时间并提高系统的整体吞吐量和可靠性。在识别出待调度的节点之后，首要步骤是筛选出在节点负载模型上计算负载 d_j 超过预设阈值 d_{th} 的节点，并将这些节点归入集合 V 中。并依据节点负载模型所承载的负载大小，对集合 V 内的节点进行排序，排序结果以负载从高到低的顺序被组织进队列 C_V 中。按照此排序逻辑，队列 C_V 的首个特征即为优先考虑进行迁移的节点，其表达式为：

$$\varpi = \arg \max_j t_j \quad (12)$$

式中： t_j 代表待迁移节点上的节点负载模型的负载的最大值。

综上所述，该算法的整体设计流程如图2所示。

输入：确定各物理节点的核心资源指标，包括CPU处理能力 S_{CPU_i} 、内存占用 S_{use_i} 及网络带宽 S_{band_i} 。同时，明确各节点上所有节点所承担的计算任务量集合，表示为 $S=(S_1, S_2, \dots, S_n)$ 。

(1) 性能评估与负载计算：依据公式(9)评估各节点的综合性能量化值。随后，利用公式(10)计算各节点的相对负载状况，并基于这些数据，通过公式(11)计算出系统的并行计算熵，以衡量系统的负载均衡度。

(2) 熵值判断：检查计算出的并行计算熵是否超过预设阈值 d_{th} 。若超过，则表明当前系统负载已相对均衡，算法结束，否则继续执行迁移优化流程。

(3) 源节点选择：根据特定策略，从待迁移源节点集合 C_s 中选取当前优先处理的源节点 $node_e$ 。

(4) 节点负载模型选择：从为 $node_e$ 准备的待迁移节点负载模型的队列 C_V 中，选择队首的节点作为当前调度节点。

(5) 目标节点确定：从潜在的目标节点队列 C_j 中选取最适合接收节点 $node_e$ 。

(6) 执行调度：将此次调度操作记录到迁移决策集 S 中，并实际执行调度操作，即将节点从 $node_e$ 迁移至 $node_e$ 。

(7) 迭代优化：调度完成后，重新评估所有节点的性能量化值、相对负载以及系统的并行计算熵^[16]，以反映迁移操作对系统状态的影响。之后回到步骤(2)，重复上述过程，直至系统达到预定的负载均衡状态，或无法再通过迁移进一步优化^[17]。

通过上述步骤，节点资源得以高效利用，调度决策更加科学合理，有助于提升整体系统的性能和稳定性。实现节点

负载的均衡分配需要综合考虑多种因素，包括系统的架构、节点的性能、请求的性质以及业务的实际需求等。在实际应用中，可以根据具体情况选择合适的负载均衡策略和技术，以实现资源的优化配置和系统的稳定运行。

2 实验测试与分析

2.1 实验准备

为验证本文设计的高性能并行计算中节点负载均衡分配算法具有较高的可行性，现进行实验测试，以验证该算法的有效性。首先选用Matlab软件搭建一实验仿真平台，其硬件配置如表1所示。

表1 硬件配置

参数	数据
操作系统	Linux CentOS7
CPU 主频	3.90 GHz
虚拟机	VMware Workstation2.0
内存	16 GB
云计算平台	Hadoop2.4.1

其次，在本次实验测试中，设定了一个实验环境，其中网络节点总数被设定为348个。为了模拟信号传输过程中的复杂情况，引入了多径分量，并设定其相位偏移量为0.3 rad/s。此外，为优化实验质量，应用信道均衡技术，其均衡系数被精确调整为0.85。同时，针对信号传输中的时延问题，设置时延均衡调节机制，其调节系数为0.36。在信号编码方面，设定码元速率为1 kBaud，以确保数据传输的效率与稳定性。基于上述设计的仿真参数，进行了本次实验测试，以评估系统性能并进一步优化相关配置。

2.2 实验结果与分析

基于上述实验准备，本次测试选用文献1、2方法与本文方法进行对比验证，同时，实验在固定时间内检测三种方法的节点的负载均衡分配数量，其测试结果如表2所示。

表2 测试结果

测试时间/h	本文方法的节点分配数量/个	方法1的节点分配数量/个	方法2的节点分配数量/个
1	341	128	214
2	329	163	230
3	308	157	258
4	317	139	224
5	336	185	269
6	314	147	213
7	322	162	207

根据上述实验结果可以看出,本文设计的高性能并行计算中节点负载均衡分配算法能够在固定测试时间内最大程度上对节点负载进行均衡分配,而另外两种方法的分配数量较少,因此可以证明本文方法的有效性。

综上所述,通过全面而深入地分析高性能并行计算领域中节点负载均衡分配算法的实验结果,可以更加精准地把握不同算法在不同场景下的适用性和优势。并结合具体的系统需求和场景特点,可以选择或设计合适的负载均衡算法,并通过引入智能预测机制、优化任务划分策略、增强系统可扩展性、考虑网络拓扑结构、加强监控和日志记录,以及引入容错机制等措施,来不断改进和优化算法的性能和效果。

3 结语

在探讨高性能并行计算中节点负载均衡分配算法的过程中,深入理解负载均衡在提升系统性能和效率方面的重要性。负载均衡不仅仅是简单地将任务分配给各个节点,更是要在保证系统资源高效利用的同时,实现各节点之间负载的均衡,从而避免“忙者愈忙,闲者愈闲”的现象。通过对多种负载均衡算法的分析,认识到每种算法都有其独特的优势和适用场景。静态负载均衡算法在程序执行前便进行任务分配,虽然简单易行,但难以应对动态变化的负载情况。而动态负载均衡算法则能够根据实时负载情况进行动态调整,更加灵活高效。其中随机算法、轮转算法、最小负载算法以及自适应负载均衡算法等,各自在解决负载均衡问题上展现了不同的策略和效果。特别是自适应负载均衡算法,它结合了静态和动态负载均衡的优点,通过动态调整任务分配策略来适应系统负载的变化,为系统提供了更高的灵活性和稳定性。这种算法的应用,不仅提高了系统的并行效率,还使得系统能够更好地应对复杂多变的计算任务。

参考文献:

- [1] 沈小渝,李晗,李红映.基于自适应遗传算法的无线激光通信网络负载均衡成簇算法[J].激光杂志,2023,44(3):205-209.
- [2] 胡逸飞,包梓群,包晓安.一种基于改进遗传算法的动态负载均衡算法[J].电子科技,2023,36(9):79-85.
- [3] 杜泰,崔鑫,陈向效,等.基于模糊离散粒子群的数据中心负载均衡研究[J].山东理工大学学报(自然科学版),2024,38(5):61-68.
- [4] 魏德宾,王英杰,梁超.卫星网络混合负载均衡策略下的多径流量分配算法[J].计算机工程与设计,2024,45(6):1654-1660.
- [5] 宋继勤,周春雷,沈子奇,等.基于TCN-GRU的Handle

标识解析系统负载均衡算法[J].福建师范大学学报(自然科学版),2024,40(2):64-73.

- [6] 滕飞,刘洋,曹芙.基于节点实时负载的开源大数据负载均衡优化算法[J].吉林大学学报(信息科学版),2023,41(6):1106-1111.
- [7] 郭文磊.复杂建筑群物联网非均匀分布节点随机负载均衡性优化模型[J].电信科学,2023,39(7):139-148.
- [8] 韩承浩,陈乃金,胡宇杨,等.负载均衡的2D Mesh单节点故障容错路由算法[J].长春理工大学学报(自然科学版),2023,46(2):128-135.
- [9] 许世辉,黄秀彬,安业腾,等.基于多Agent技术的多运营终端自适应负载均衡算法[J].计算技术与自动化,2022,41(3):77-81.
- [10] 李静,张英争,郭静,等.无人机通信网络非等间距节点负载部署方法[J].河南机电高等专科学校学报,2022,30(2):17-21.
- [11] 高洁,马杰,杨丽新.基于非线性容量负载模型下电网的级联故障分析[J].科学技术与工程,2022,22(9):3601-3606.
- [12] 吴健,吴奎华,崔灿,等.分布式新能源配电网输出线路负载诊断模型分析[J].沈阳工业大学学报,2023,45(3):259-263.
- [13] 周妃,周静.基于图谱特征提取的宽带城域光纤网络异常识别方法[J].激光杂志,2022,43(5):116-120.
- [14] 徐聪,叶钧超,黄尧,等.基于联合权重超图划分的SNN负载均衡方法[J].计算机应用研究,2023,40(7):2130-2137.
- [15] 苏玉钊,孙恩昌,杨睿哲,等.面向Kafka共识消息传输的负载均衡算法[J].高技术通讯,2023,33(1):42-49.
- [16] 谭印,苏雯洁.基于局部信息熵的计算机网络高维数据离群点检测系统[J].现代电子技术,2024,47(10):91-95.
- [17] 王正来,关胜晓.基于改进积分梯度的黑盒迁移攻击算法[J].计算机工程与应用,2024,60(9):309-316.

【作者简介】

田彦明(1978—),男,辽宁兴城人,本科,研究方向:高级工程师、航空信息化。

侯永生(1981—),男,辽宁沈阳人,本科,研究方向:高级工程师、网络安全、数据安全。

高超(1984—),男,辽宁昌图人,本科,高级工程师,研究方向:网络安全、数据安全、航空信息化。

(收稿日期:2024-07-30)