



(21) 申请号 202411572322.3

(22) 申请日 2024.11.06

(65) 同一申请的已公布的文献号

申请公布号 CN 119088572 A

(43) 申请公布日 2024.12.06

(73) 专利权人 四川万物纵横科技股份有限公司

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 毛开福 刘瑶 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

专利代理师 安宁

(51) Int. Cl.

G06F 9/50 (2006.01)

G06N 3/126 (2023.01)

(56) 对比文件

马景奕;隋兵;舒万能.“基于Min-Min遗传算法的网格任务调度方法”.《计算机工程与应用》.2008,(23),第103页左栏-104页左栏.

审查员 王璐

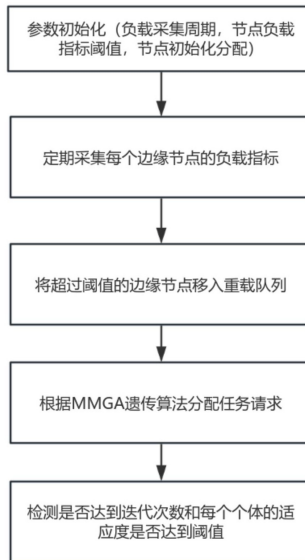
权利要求书3页 说明书7页 附图3页

(54) 发明名称

一种基于改进的遗传算法负载均衡的方法

(57) 摘要

本发明公开了一种基于改进的遗传算法负载均衡的方法,属于集群服务器技术领域,包括以下步骤:参数初始化;定期采集每个边缘节点的负载指标;将负载指标超过阈值的边缘节点移入重载队列;根据遗传算法与优化后的Min-Min算法进行分配任务请求;检测是否达到迭代次数和每个个体的适应度是否达到阈值;本发明通过预设指标的权重分析,对Min-Min算法进行优化,并将Min-Min算法与遗传算法相结合,以求得最优的分配方式,可生成任务处理总耗时最短的生产调度问题,并利用具有高效、并行、全局搜索特点的遗传算法来求此生产调度问题的最优解,此最优解即为负载分配方案。



1.一种基于改进的遗传算法负载均衡的方法,其特征在于,包括以下步骤:

S1:按照预设的时间周期,定期采集集群中位于适载队列中的各个边缘节点的负载指标,并将负载指标超过预设越界阈值的边缘节点移入重载队列;

S2:根据遗传算法,基于适载队列,对待解决问题进行编码,即每一种任务请求分配方案编码为一个染色体结构的个体;

S3:确认群体规模ksize、停止迭代次数Tm、交叉概率Pc、变异概率Pm、各个个体的选择概率Ps和各个个体的适应度函数值f;

S4:根据Min-Min算法,选取ksize个解作为初始种群 P_0 , $P_0 = \{P_1 \cdots P_{ksize}\}$,将初始种群 P_0 作为当前群体并进入下一步;

S5:通过层次分析法对当前群体的负载指标进行权重分析,并进一步计算评价当前群体中各个个体的适应度函数值f,第i个个体的适应度函数值f具体表达为:

$$f(i) = \frac{1}{makespan(i) * \sigma(i)};$$

其中,makespan(i)表示第i个个体所有任务都执行完成的时间跨度;

$$makespan(i) = \max_{j=1}^m (Min_{ji});$$

Min为任务完成时间矩阵,Min为 $m \times n$ 矩阵,每一行代表同一任务请求在集群的n个不同服务节点上的完成时间,每一列表示在集群中同一服务节点上m个不同任务请求的完成时间,其中 Min_{ji} 表示第j个任务在第i个服务节点上的完成时间;

σ 表示为服务节点集群负载的标准差:

$$\sigma(i) = \sqrt{\frac{\sum_{i=1}^n (L(S_i) - \frac{\sum_{i=1}^n L(S_i)}{n})^2}{n}};$$

其中 S_i 代表集群中第i个节点,设置4个评价指标,包括服务节点的CPU利用率 Cu_i 、磁盘利用率 Mu_i 、内存利用率 Bu_i 、带宽利用率 Tu_i ,负载指标 $L(S_i) = \alpha^2 Cu_i + \beta^2 Mu_i + \gamma^2 Bu_i + \delta^2 Tu_i$,其中, $\alpha^2 + \beta^2 + \gamma^2 + \delta^2 = 1$, Cu_i 、 Mu_i 、 Bu_i 、 Tu_i 分别表示第i个节点的CPU利用率、磁盘利用率、内存利用率和带宽利用率, α^2 表示CPU利用率权重, β^2 表示磁盘利用率权重, γ^2 表示内存利用率权重, δ^2 表示带宽利用率权重;

其中权重大小根据层次分析法来计算:

构造的判断矩阵为 4×4 判断矩阵;

将评价指标两两进行对比,可得到判断矩阵为:

$$A = \begin{bmatrix} 1 & 1 & 4 & \frac{1}{3} \\ 1 & 1 & 4 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{5} \\ 3 & 3 & 5 & 1 \end{bmatrix};$$

计算每个判断指标的权重系数,求出矩阵A最大特征值所对应的特征向量为:

$(-0.35088504, -0.35088504, -0.11223483, -0.86090808)T$;

对特征向量进行归一化处理,得:

$(0.21, 0.21, 0.07, 0.51)T$;

可知CPU利用率 Cu 的指标权重 α_2 为0.21,磁盘利用率 Mu 的权重 β_2 为0.21,内存利用率 Bu 的权重 γ_2 为0.07,带宽利用率 Tu 的指标权重 δ_2 为0.51;

则最终负载指标为 $L(S_i) = 0.21Cu_i + 0.21Mu_i + 0.07Bu_i + 0.51Tu_i$;

S6:通过各个个体的适应度函数值 f 计算当前群体中各个个体的选择概率 P_s ,选择概率 P_s 的具体计算方法为:

$$P_s(i) = \frac{f(i)}{\sum_{i=1}^{ksize} f(i)};$$

其中, $P_s(i)$ 表示当前群体中第 i 个个体的选择概率, $f(i)$ 表示当前群体中第 i 个个体的适应度函数值;

S7:进行迭代操作;生成新群体,新群体为空集,生成一个随机数 m , $m = \text{random}[0, 1]$,将当前群体中的每个个体作为父体,并进行以下条件判断:

条件1:父体的选择概率 $P_s \geq m$;

条件2:父体的选择概率 $P_s < m$,且 $m \geq P_c + P_m$;

条件3:父体的选择概率 $P_s < m$,且 $m < P_c + P_m$;

将满足条件1的父体直接插入新群体中,对满足条件2的父体两两之间执行交叉操作,将交叉操作后生成的后代插入到新群体中,对满足条件3的父体分别执行变异操作,将变异操作后生成的后代插入到新群体中;

S8:设定迭代终止条件,满足迭代终止条件则停止迭代操作并得到最终任务请求分配方案集合,不满足迭代终止条件则将新群体作为当前群体并返回S4。

2.根据权利要求1所述的一种基于改进的遗传算法负载均衡的方法,其特征在于,S1中预设的越界阈值包括边缘节点的CPU负载阈值 F_c 、磁盘负载阈值 F_d 、内存负载阈值 F_m 、带宽负载阈值 F_b 。

3.根据权利要求1所述的一种基于改进的遗传算法负载均衡的方法,其特征在于,染色体长度对应任务请求数量,染色体中各个基因位置的值代表任务请求编号。

4.根据权利要求1所述的一种基于改进的遗传算法负载均衡的方法,其特征在于,S3中, $10 \leq ksize \leq 300$, $200 \leq T_m \leq 1000$, $0.4 \leq P_c \leq 0.99$, $0.001 \leq P_m \leq 0.1$ 。

5.根据权利要求1所述的一种基于改进的遗传算法负载均衡的方法,其特征在于,S7中,交叉概率 P_c 的计算方法为:

$$P_c = \begin{cases} \frac{k1(f_{max} - f_c^{max})}{f_{max} - f_{avg}}, & f_c^{max} \geq f_{avg}; \\ k2, & f_c^{max} < f_{avg} \end{cases};$$

其中, f_{max} 表示当前种群中适应度函数值的最大值, f_{avg} 表示当前种群中适应度函数值的平均值, f_c^{max} 表示参与交叉父代中适应度函数值较大一方的适应度函数值, $k1, k2$ 为常数,其中 $k1 < k2$;

变异概率 P_m 的计算方法为:

$$P_m = \begin{cases} \frac{k_3(f_{max} - f_m)}{f_{max} - f_{avg}}, & f_m \geq f_{avg}; \\ k_4, & f_m < f_{avg} \end{cases};$$

其中, f_m 表示参与变异的父体适应度函数值, k_3, k_4 为常数, 其中 $k_3 < k_4$ 。

6. 根据权利要求1所述的一种基于改进的遗传算法负载均衡的方法, 其特征在于, S8中, 终止条件如下:

当迭代次数达到停止迭代次数 T_m , 则终止迭代, 或者所有个体的适应函数值 f 达到阈值, 则终止迭代。

一种基于改进的遗传算法负载均衡的方法

技术领域

[0001] 本发明属于集群服务器技术领域,具体涉及一种基于改进的遗传算法负载均衡的方法。

背景技术

[0002] 集群的出现使得人们可以用廉价的计算资源完成大规模的科学计算,提高系统计算速度与可靠性。集群在并行计算和分布式计算中具有巨大的潜力,而任务调度则是挖掘这些潜力的关键技术。

[0003] 在以往的研究中,已经产生了许多任务调度算法,如Min-Min算法、Max-Min算法、遗传算法、模拟退火算法等。

[0004] 其中Min-Min作为启发式算法中的经典算法之一,简单且容易实现,同时具有较好的时间跨度,Min-Min算法总是先执行具有最短完成时间的作业,有着算法思路简单、总完成时间短的特点;

[0005] 但是该算法考虑的因素较少,不能确保网络资源的负载均衡。

[0006] 遗传算法也作为一种著名的元启发式算法,模拟了进化和遗传原理。遗传从全局解空间中搜索最优解,通常可以获得较好的解决方案;

[0007] 但是遗传算法容易陷入只能获取到局部最小值的困境,从而难以得到整体的最优解。

发明内容

[0008] 为解决上述背景技术中提出的问题,本发明提供一种基于改进的遗传算法负载均衡的方法,以解决Min-Min算法考虑的因素较少,不能确保网络资源的负载均衡的问题,以及遗传算法容易陷入只能获取到局部最小值的困境,从而难以得到整体的最优解。

[0009] 为实现上述目的,本发明提供如下技术方案:

[0010] 一种基于改进的遗传算法负载均衡的方法,包括以下步骤:

[0011] S1:按照预设的时间周期,定期采集集群中位于适载队列中的各个边缘节点的负载指标,并将负载指标超过预设越界阈值的边缘节点移入重载队列;

[0012] S2:根据遗传算法,基于适载队列,对待解决问题进行编码,即每一种任务请求分配方案编码为一个染色体结构的个体;

[0013] S3:确认群体规模ksize、停止迭代次数Tm、交叉概率Pc、变异概率Pm、各个个体的选择概率Ps和各个个体的适应度函数值f;

[0014] S4:根据Min-Min算法,选取ksize个解作为初始种群 P_0 , $P_0 = \{P_1 \cdots P_{ksize}\}$,将初始种群 P_0 作为当前群体并进入下一步;

[0015] S5:通过层次分析法对当前群体的负载指标进行权重分析,并进一步计算评价当前群体中各个个体的适应度函数值f,第i个个体的适应度函数值f具体表达为:

$$[0016] \quad f(i) = \frac{1}{\text{makespan}(i) \cdot \sigma(i)};$$

[0017] 其中, makespan(i) 表示第i个个体所有任务都执行完成的时间跨度;

$$[0018] \quad \text{makespan}(i) = \max_{j=1}^m (\text{Min}_{ji});$$

[0019] Min为任务完成时间矩阵, Min为m*n矩阵, 每一行代表同一任务请求在集群的n个不同服务节点上的完成时间, 每一列表示在集群中同一服务节点上m个不同任务请求的完成时间, 其中Min_{ji}表示第j个任务在第i个服务节点上的完成时间;

[0020] σ 表示为服务节点集群负载的标准差:

$$[0021] \quad \sigma(i) = \sqrt{\frac{\sum_{i=1}^n (L(S_i) - \frac{\sum_{i=1}^n L(S_i)}{n})^2}{n}};$$

[0022] 其中S_i代表集群中第i个节点, 设置4个评价指标, 包括服务节点的CPU利用率Cu、磁盘利用率Mu、内存利用率Bu、带宽利用率Tu, 负载指标

$L(S_i) = \alpha^2 Cu_i + \beta^2 Mu_i + \gamma^2 Bu_i + \delta^2 Tu_i$, 其中, $\alpha^2 + \beta^2 + \gamma^2 + \delta^2 = 1$, Cu_i、Mu_i、Bu_i、Tu_i分别表示第i个节点的CPU利用率、磁盘利用率、内存利用率和带宽利用率, α^2 表示CPU利用率权重, β^2 表示磁盘利用率权重, γ^2 表示内存利用率权重, δ^2 表示带宽利用率权重;

[0023] 其中权重大小根据层次分析法来计算;

[0024] 构造的判断矩阵为4×4判断矩阵;

[0025] 将评价指标两两进行对比, 可得到判断矩阵为:

$$[0026] \quad A = \begin{bmatrix} 1 & 1 & 4 & \frac{1}{3} \\ \frac{1}{1} & 1 & 4 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{5} \\ 3 & 3 & 5 & 1 \end{bmatrix};$$

[0027] 计算每个判断指标的权重系数, 求出矩阵A最大特征值所对应的特征向量为:

[0028] (-0.35088504, -0.35088504, -0.11223483, -0.86090808)T;

[0029] 对特征向量进行归一化处理, 得:

[0030] (0.21, 0.21, 0.07, 0.51)T;

[0031] 可知CPU利用率Cu的指标权重 α^2 为0.21, 磁盘利用率Mu的权重 β^2 为0.21, 内存利用率Bu的权重 γ^2 为0.07, 带宽利用率Tu的指标权重 δ^2 为0.51;

[0032] 则最终负载指标为 $L(S_i) = 0.21Cu_i + 0.21Mu_i + 0.07Bu_i + 0.51Tu_i$;

[0033] S6: 通过各个个体的适应度函数值f计算当前群体中各个个体的选择概率Ps, 选择概率Ps的具体计算方法为:

$$[0034] \quad P_s(i) = \frac{f(i)}{\sum_{i=1}^{\text{ksize}} f(i)};$$

[0035] 其中, Ps(i)表示当前群体中第i个个体的选择概率, f(i)表示当前群体中第i个个

体的适应度函数值；

[0036] S7:进行迭代操作;生成新群体,新群体为空集,生成一个随机数 m , $m=\text{random}[0,1]$,将当前群体中的每个个体作为父体,并进行以下条件判断:

[0037] 条件1:父体的选择概率 $P_s \geq m$;

[0038] 条件2:父体的选择概率 $P_s < m$,且 $m \geq P_c + P_m$;

[0039] 条件3:父体的选择概率 $P_s < m$,且 $m < P_c + P_m$;

[0040] 将满足条件1的父体直接插入新群体中,对满足条件2的父体两两之间执行交叉操作,将交叉操作后生成的后代插入到新群体中,对满足条件3的父体分别执行变异操作,将变异操作后生成的后代插入到新群体中;

[0041] S8:设定迭代终止条件,满足迭代终止条件则停止迭代操作并得到最终任务请求分配方案集合,不满足迭代终止条件则将新群体作为当前群体并返回S4。

[0042] 优选的,S1中预设的越界阈值包括边缘节点的CPU负载阈值 F_c 、磁盘负载阈值 F_d 、内存负载阈值 F_m 、带宽负载阈值 F_b 。

[0043] 优选的,染色体长度对应任务请求数量,染色体中各个基因位置的值代表任务请求编号。

[0044] 优选的,S3中, $10 \leq ksize \leq 300$, $200 \leq T_m \leq 1000$, $0.4 \leq P_c \leq 0.99$, $0.001 \leq P_m \leq 0.1$ 。

[0045] 优选的,S7中,交叉概率 P_c 的计算方法为:

$$[0046] \quad P_c = \begin{cases} \frac{k1(f_{\max} - f_c^{\max})}{f_{\max} - f_{\text{avg}}}, & f_c^{\max} \geq f_{\text{avg}} \\ k2, & f_c^{\max} < f_{\text{avg}} \end{cases};$$

[0047] 其中, f_{\max} 表示当前种群中适应度函数值的最大值, f_{avg} 表示当前种群中适应度函数值的平均值, f_c^{\max} 表示参与交叉父代中适应度函数值较大一方的适应度函数值, $k1$, $k2$ 为常数,其中 $k1 < k2$;

[0048] 变异概率 P_m 的计算方法为:

$$[0049] \quad P_m = \begin{cases} \frac{k3(f_{\max} - f_m)}{f_{\max} - f_{\text{avg}}}, & f_m \geq f_{\text{avg}} \\ k4, & f_m < f_{\text{avg}} \end{cases};$$

[0050] 其中, f_m 表示参与变异的父体适应度函数值, $k3$, $k4$ 为常数,其中 $k3 < k4$ 。

[0051] 优选的,S8中,终止条件如下:

[0052] 当迭代次数达到停止迭代次数 T_m ,则终止迭代,或者所有个体的适应函数值 f 达到阈值,则终止迭代。

[0053] 与现有技术相比,本发明的有益效果是:

[0054] 本发明通过预设指标的权重分析,对Min-Min算法进行优化,并将Min-Min算法与遗传算法相结合,以求得最优的分配方式,可生成任务处理总耗时最短的生产调度问题,并利用具有高效、并行、全局搜索特点的遗传算法来求此生产调度问题的最优解,此最优解即为负载分配方案。

附图说明

- [0055] 图1为本申请的流程示意图；
 [0056] 图2为本申请的任务-服务节点编码图；
 [0057] 图3为本申请的层次结构示意图；
 [0058] 图4为本申请的遗传算法流程图。

具体实施方式

[0059] 为便于本领域技术人员理解本发明的技术内容,以下结合附图和具体的实例对本发明作进一步地详细说明。应当理解,此处所描述的具体实例仅用以解释本发明,并不用于限定本发明。

[0060] 实施例1

[0061] 如图1、图2所示,一种基于改进的遗传算法负载均衡的方法,包括以下步骤:

[0062] S1:初始化参数,按照预设的时间周期,定期采集集群中各个边缘节点的负载指标,并将负载指标超过预设越界阈值的边缘节点移入重载队列,预设的越界阈值包括边缘节点的CPU负载阈值Fc、磁盘负载阈值Fd、内存负载阈值Fm、带宽负载阈值Fb;

[0063] 当边缘节点的任何一项资源负载超过设定的越界阈值,表示该边缘节点工作状态不太健康,将边缘节点移入重载队列,可保证当前种群的健康状态,提高种群利用效率。

[0064] S2:根据遗传算法,将基于适载队列,对待解决问题进行编码,即每一种任务请求分配方案编码为一个染色体结构的个体,染色体长度对应任务请求数量,各个基因位置的值代表任务请求编号,服务节点编号用各个基因位置中的值来表示。假设将m个任务请求(J1.....Jm)分配给n(R1,....Rn)个服务节点如图4所示。

[0065] 用矩阵结构解释染色体结构即表示为任务分配矩阵 $\text{Min_S}_{m \times n}$,其元素 $\text{Min_S}_{ji}=1$,表示任务Jj分配到服务器节点Ri上,否则, $\text{Min_S}_{ji}=0$;

[0066] 下列为把5个请求分配给3个服务器节点的一种方案的染色体编码的示例,把5个请求分配到3个服务器节点上,请求1被分到边缘节点2上,请求2被分到边缘节点3上,请求3被分到边缘节点1上,请求4被分到边缘节点2上,请求5被分到边缘节点1上,遗传算法中每条染色体代表一种分配方案,进行染色体编码如下矩阵所示:

$$[0067] \quad \text{Min_S} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}。$$

[0068] S3:确认群体规模ksize、停止迭代次数Tm、交叉概率Pc、变异概率Pm,其中 $10 \leq ksize \leq 300$, $200 \leq Tm \leq 1000$, $0.4 \leq Pc \leq 0.99$, $0.001 \leq Pm \leq 0.1$ 。

[0069] S4:根据Min-Min算法,设定个体数量的集合为初始种群Po, $Po = \{P1 \dots P_{ksize}\}$,将初始种群Po作为当前群体并进入下一步。

[0070] S5:通过预设指标的权重分析,计算当前群体中各个个体的适应度函数值f;

[0071] 第i个个体的适应度函数值f具体表达为:

$$[0072] \quad f(i) = \frac{1}{\text{makespan}(i) \cdot \sigma(i)};$$

[0073] 其中,makespan(i)表示第i个个体所有任务都执行完成的时间跨度;

[0074] $\text{makespan}(i) = \max_{j=1}^m(\text{Min}_{ji})$;

[0075] Min为任务完成时间矩阵,Min为m*n矩阵,每一行代表同一任务请求在集群的n个不同服务节点上的完成时间,每一列表示在集群中同一服务节点上m个不同任务请求的完成时间,其中 Min_{ji} 表示第j个任务在第i个服务节点上的完成时间。

[0076] 其中, σ 表示为服务节点集群负载的标准差, $\sigma(i) = \sqrt{\frac{\sum_{i=1}^n (L(S_i) - \frac{\sum_{i=1}^n L(S_i)}{n})^2}{n}}$;

[0077] 最优跨度,即使任务的总执行时间(makespan)最小。故选用任务的总执行时间函数结合服务器节点的负载作为遗传算法的适应度函数。

[0078] 令 S_i 代表集群中第i个节点,设置4个评价指标,包括边缘节点的CPU利用率 Cu 、磁盘利用率 Mu 、内存利用率 Bu 、带宽利用率 Tu ,负载指标

$L(S_i) = \alpha^2 Cu_i + \beta^2 Mu_i + \gamma^2 Bu_i + \delta^2 Tu_i$,其中, $\alpha^2 + \beta^2 + \gamma^2 + \delta^2 = 1$, Cu_i 、 Mu_i 、 Bu_i 、 Tu_i 分别表示第i个节点的CPU利用率、磁盘利用率、内存利用率和带宽利用率, α^2 表示CPU利用率权重, β^2 表示磁盘利用率权重, γ^2 表示内存利用率权重, δ^2 表示带宽利用率权重;

[0079] 其中权重设定由层次分析法决定,如图3所示建立层次结构,层次分为目标层、准则层、方案层;

[0080] (1) 构造判断矩阵。其中准则层的评价指标有4个,则构造的判断矩阵为4*4;

[0081] (2) 将准则层的评价指标两两进行对比,可得到判断矩阵为:

[0082]
$$A = \begin{bmatrix} 1 & 1 & 4 & \frac{1}{3} \\ \frac{1}{1} & \frac{1}{1} & 4 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{5} \\ 3 & 3 & 5 & 1 \end{bmatrix};$$

[0083] (3) 对得到的判断矩阵进行一致性检验,通过计算一致性比例CR来验证求得 $CR < 0.1$,此判断矩阵合理;

[0084] (4) 计算每个指标的权重系数。求出矩阵T最大特征值所对应的特征向量为 $(-0.35088504, -0.35088504, -0.11223483, -0.86090808)T$,并对其进行归一化处理 $(0.21, 0.21, 0.07, 0.51)T$ 。可知CPU利用率 Cu 的指标权重 α^2 为0.21,磁盘利用率 Mu 的权重 β^2 为0.21,内存利用率 Bu 的权重 γ^2 为0.07,带宽利用率 Tu 的指标权重 δ^2 为0.51;

[0085] 则最终负载指标为 $L(S_i) = 0.21Cu_i + 0.21Mu_i + 0.07Bu_i + 0.51Tu_i$ 。

[0086] S6:通过各个个体的适应函数值f计算当前群体中各个个体的选择概率 Ps ;

[0087] 选择概率 Ps 的具体计算方法为:

[0088]
$$Ps(i) = \frac{f(i)}{\sum_{i=1}^{ksize} f(i)};$$

[0089] 其中, $Ps(i)$ 表示当前群体中第i个个体的选择概率, $f(i)$ 表示当前群体中第i个个体的适应度函数值。

[0090] S7:进行迭代操作;生成新群体,新群体为空集,生成一个随机数 m , $m=\text{random}[0,1]$,将当前群体中的每个个体作为父体,并进行以下条件判断:

[0091] 条件1:父体的选择概率 $P_s(i) \geq m$;

[0092] 条件2:父体的选择概率 $P_s(i) < m$,且 $m \geq P_c + P_m$;

[0093] 条件3:父体的选择概率 $P_s(i) < m$,且 $m < P_c + P_m$;

[0094] 将满足条件1的父体直接插入新群体中,对满足条件2的父体两两之间执行交叉操作,将交叉操作后生成的后代插入到新群体中,对满足条件3的父体分别执行变异操作,将变异操作后生成的后代插入到新群体中;

[0095] 其中,交叉概率 P_c 的计算方法为:

$$[0096] \quad P_c = \begin{cases} \frac{k1(f_{\max} - f_c^{\max})}{f_{\max} - f_{\text{avg}}}, & f_c^{\max} \geq f_{\text{avg}}; \\ k2, & f_c^{\max} < f_{\text{avg}} \end{cases};$$

[0097] 其中, f_{\max} 表示当前种群中适应度函数值的最大值, f_{avg} 表示当前种群中适应度函数值的平均值, f_c^{\max} 表示参与交叉父代中适应度函数值较大一方的适应度函数值, $k1$, $k2$ 为常数,其中 $k1 < k2$;

[0098] 变异概率 P_m 的计算方法为:

$$[0099] \quad P_m = \begin{cases} \frac{k3(f_{\max} - f_m)}{f_{\max} - f_{\text{avg}}}, & f_m \geq f_{\text{avg}}; \\ k4, & f_m < f_{\text{avg}} \end{cases};$$

[0100] 其中, f_m 表示参与变异的父体适应度函数值, $k3$, $k4$ 为常数,其中 $k3 < k4$ 。

[0101] S8:不能让遗传算法一直迭代下去,这样既浪费边缘节点性能,又不能及时为请求分配边缘节点,因此我们要设定迭代终止条件,终止条件如下:

[0102] 当迭代次数达到停止迭代次数 T_m ,则终止迭代,或者所有个体的适应函数值 f 达到阈值,则终止迭代。

[0103] S9:最后为了评估MMGA算法的性能,利用整个集群的平均负载来衡量集群资源利用率,平均负载越大,集群资源利用率越高,假设集群为 n , S_i 为第 i 个节点平均负载计算公式如下:

$$[0104] \quad \text{Levl} = \sum_{i=1}^n L(S_i);$$

[0105] 利用整个集群负载的标准差来衡量整个集群的负载均衡程度。标准差越小表示各个服务器节点的负载相差越小,整个集群负载均衡程度越高。标准差 σ 的计算方法如下所示:

$$[0106] \quad \sigma = \sqrt{\frac{\sum_{i=1}^n (L(S_i) - \text{Levl})^2}{n}};$$

[0107] 平均负载越高,集群资源利用率越高。标准差越小各个服务器节点负载分配越均匀,整个集群负载均衡程度越高。

[0108] 在本实施例中,本申请相较于Min-Min算法存在仅考虑负载均衡设备与服务器间的响应时间而不是客户端与服务器间的响应时间,更多只反映网络状况而没有很好反映设

备性能等问题。本发明本文提出将服务节点的cpu利用率、磁盘利用率,带宽利用率等指标与内存利用率进行加权求和,计算出一个综合指标,再以这个综合指标来衡量每个节点的负载来对Min-Min算法进行优化;并将Min-Min算法与遗传算法相结合。

[0109] 本申请将负载均衡调度问题看作m个不同复杂度的请求任务任意分配给n个性能不等的服务节点处理,以求得最优的分配方式,使得任务处理总耗时最短的生产调度问题。并利用具有高效、并行、全局搜索特点的遗传算法来求此生产调度问题的最优解,此最优解即为我们的负载分配方案。

[0110] 本申请相较于普通权重是人为分配的,主观性太强,得到的综合负载无法客观准确的衡量每个节点负载压力。本发明选取层次分析法,将所要分析的问题层次化,将问题分解成不同的组成因素,并按照因素间的相互关系及隶属关系,将因素按不同层次聚集组合,形成一个多层分析结构模型。

[0111] 本申请采用集群的负载的平均值和标准差来评估MMGA算法性能。

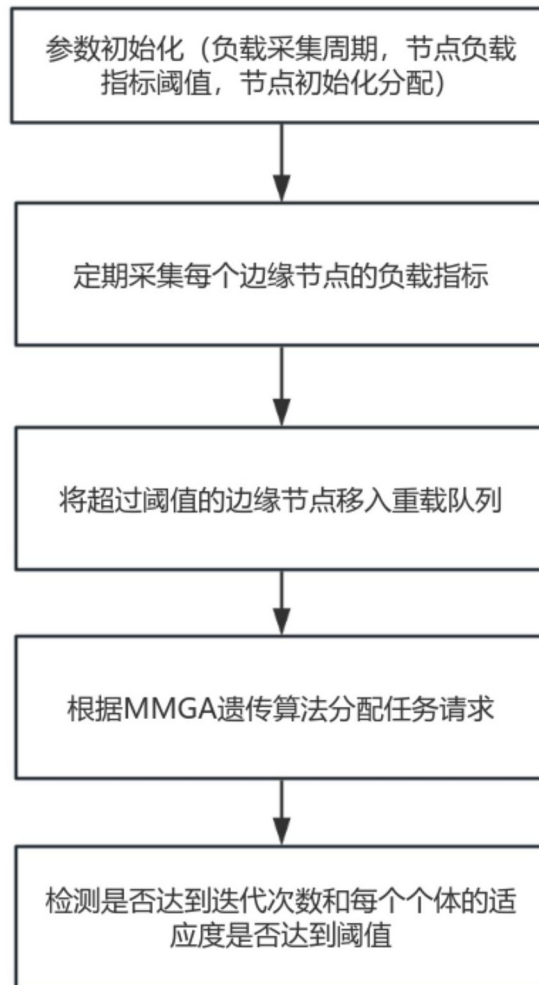


图 1

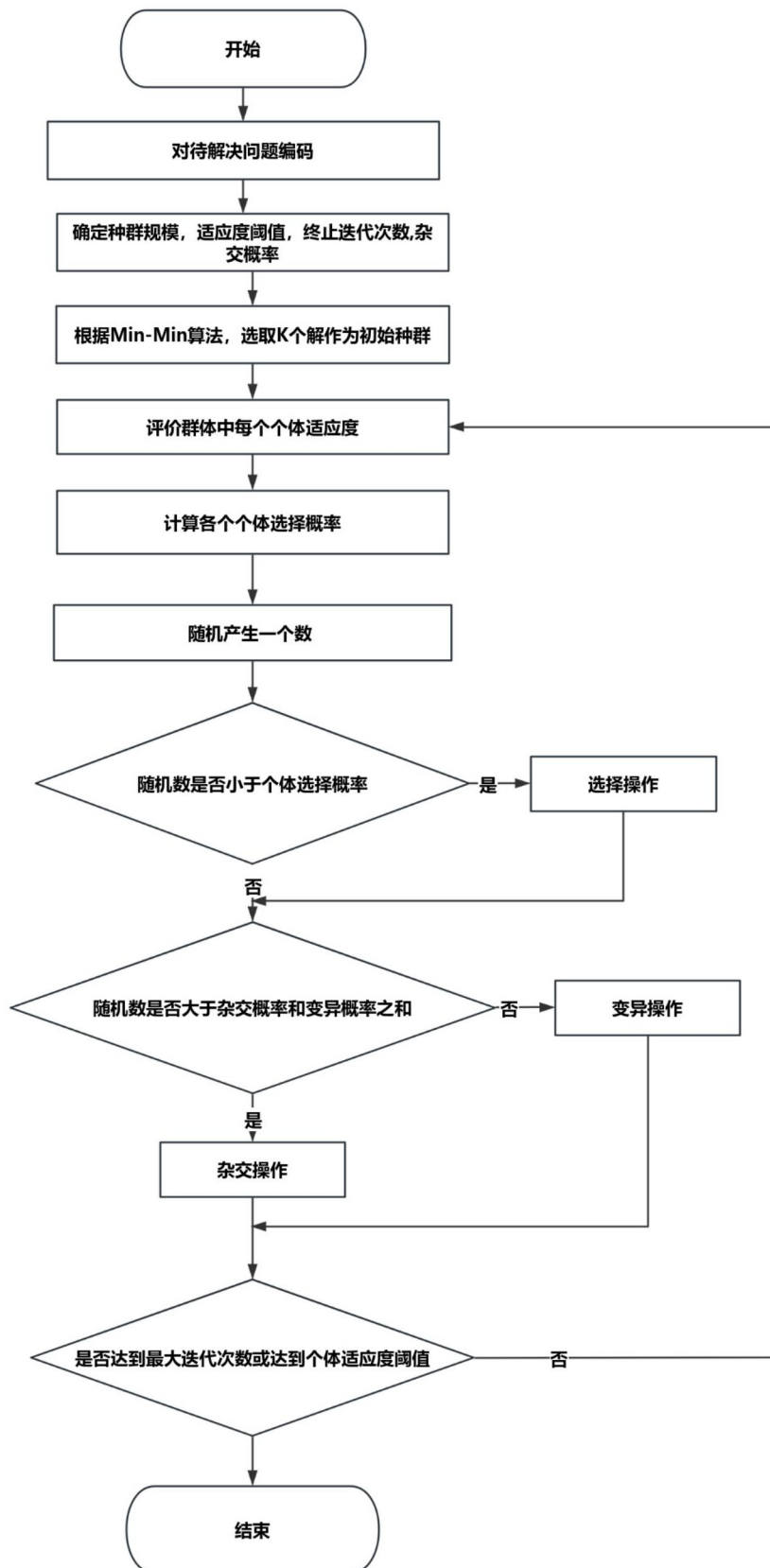


图 2

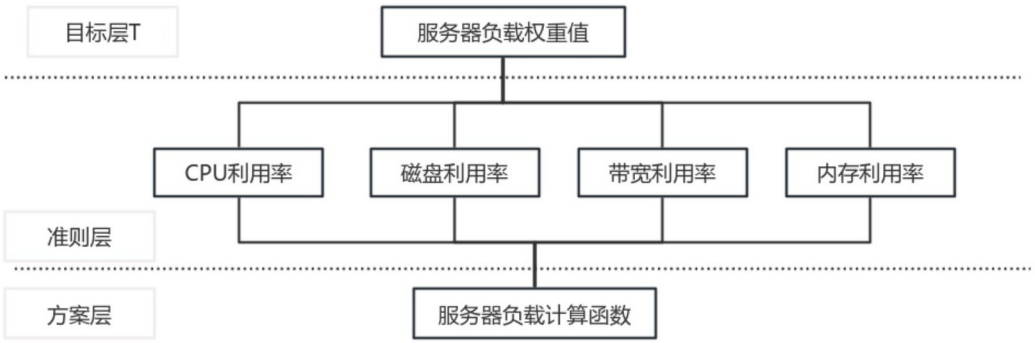


图 3



图 4