(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103677957 A
(43)申请公布日 2014.03.26

(21)申请号 201310687502.1

(22)申请日 2013.12.13

(71)申请人 重庆邮电大学
地址 400065 重庆市南岸区黄桷垭崇文路2号

(72)发明人 李鸿健 崔晟圆 唐红 豆育升

(74)专利代理机构 重庆市恒信知识产权代理有限公司 50102
代理人 刘小红

(51)Int.Cl.
G06F 9/455(2006.01)
G06N 3/00(2006.01)

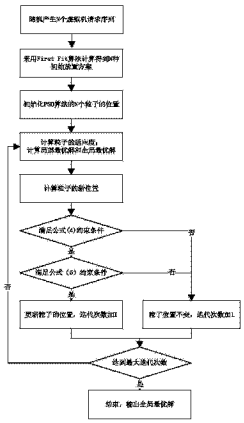
权利要求书1页 说明书8页 附图1页

(54)发明名称

云数据中心基于多资源的高能效虚拟机放置方法

(57)摘要

本发明公开了一种云数据中心基于多资源的高能效虚拟机放置方法，包括步骤：建立多资源的能效模型；采用首次适应算法生成粒子，定义粒子个体最优解和全局最优解的信任度指导粒子进化；根据多资源能效模型定义粒子群的适应度函数并以此来评价粒子。本发明方法能够有效降低云计算数据中心的虚拟机的放置的能耗，系统资源得到合理分配。



CN 103677957 A

1. 一种云数据中心基于多资源的高能效虚拟机放置方法，其特征在于，包括以下步骤：

101、获取云数据中心中多资源种类数量 d，云数据中心物理节点的数量 n 及实际利用率 u_j^i，并设定能效最佳利用率 $ubest_j^i$，其中 u_j^i 表示物理结点 i 上第 j 种资源的实际利用率，$ubest_j^i$ 表示物理结点 i 上第 j 种资源的能效最佳利用率，建立多资源能效模型，如式（1）所示：

$$\delta = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (u_j^i - ubest_j^i)^2} \qquad (1);$$

102、随机生成 N 个虚拟机请求序列，采用首次适应 First Fit 算法得到 N 种虚拟机的初始放置序列，即 N 个粒子，从而构成初始种群；

103、根据步骤 101 中得到的多资源能效模型，$\delta = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (u_j^i - ubest_j^i)^2}$，构建粒子的适应度函数 f（δ），并根据粒子群优化算法计算出适应度 f（δ）、局部最优解及全局最优解，得出粒子的新位置；

104、根据步骤 103 中得到的粒子新位置，判断是否满足 $\sum_h x_h^j = 1, \forall j$ （4）及

$$\sum_j r_j^{CPU} * x_h^j \le c_h^{CPU},\ \sum_j r_j^{RAM} * x_h^j \le c_h^{RAM},\ \sum_j r_j^{BW} * x_h^j \le c_h^{BW},\ \sum_j r_j^{DISK} * x_h^j \le c_h^{DISK} \qquad (5)$$

式（4）中的 x_h^j 表示虚拟机 j 是否放置在物理结点 h，当虚拟机 j 放置在物理结点 h 上，x_h^j 为 1，否则为 0；表示每个虚拟机只能放在一个物理结点上；式（5）表示多个虚拟机放置在物理结点 h 时，虚拟机资源不能超过物理结点 h 的资源总量，其中，r_j^{CPU}、r_j^{RAM}、r_j^{BW}、r_j^{DISK} 分别表示虚拟机 j 所需的 CPU、内存、带宽和磁盘的容量，c_h^{CPU}、c_h^{RAM}、c_h^{BW}、c_h^{DISK} 分别表示物理结点 h 的 CPU、内存、带宽、磁盘的容量；

105、若满足步骤 104 中的式（4）和式（5），则更新粒子的位置，迭代次数加 1；若不满足式（4）和式（5），则粒子位置不变，迭代次数加 1；当迭代次数≥最大迭代次数 N3 时，迭代结束并输出全局最优解，根据该全局最优解所对应的物理节点位置设置虚拟机，完成虚拟机的放置。

2. 根据权利要求 1 所述的云数据中心基于多资源的高能效虚拟机放置方法，其特征在于：步骤 101 中云数据中心的多资源种类包括 CPU、内存、带宽及磁盘。

3. 根据权利要求 1 所述的云数据中心基于多资源的高能效虚拟机放置方法，其特征在于：步骤 101 中 CPU 和磁盘的最佳能效利用率 $ubest_j^i$ 分别为 0.7 及 0.5。

云数据中心基于多资源的高能效虚拟机放置方法

技术领域

[0001] 本发明涉及云计算领域，具体是一种基于在云数据中心结合粒子群算法以节能为目标的虚拟机放置方法。

背景技术

[0002] 高能耗是云数据中心资源管理面临的一大挑战，随着数据中心规模的不断扩大，高能耗问题更加突出。例如，Google 数据中心产生的能耗相当于一个小型城市的总能耗。数据中心的高能耗不仅造成电能的浪费，系统运行的不稳定，同时对环境也造成不良的影响。造成云数据中心高能耗的主要原因有两个方面：一方面是随着用户数量的增加，数据中心基础设施建设大幅度增加，另一方面是系统资源分配不合理。研究高效的系统资源分配方法，可减少云数据中心的系统能耗，并使资源利用朝可持续方向发展。

[0003] 研究虚拟机的放置方法，定义虚拟机和物理结点之间的映射关系对云数据中心中资源的合理分配具有至关重要的作用，同时对系统的能耗、性能和资源利用率也产生重要的影响。虚拟机放置问题可以描述为向量装箱问题：装入的物品是运行中的虚拟机，箱子为物理结点，虚拟机的资源需求和物理结点的资源数量分别表示物品和箱子的大小。虚拟机和物理结点的资源包括 CPU、内存、带宽、磁盘等，资源的种类即向量的维度。对于 M 个物理结点，N 个虚拟机的云数据中心，将虚拟机部署到物理结点的解空间为 M^N，属于 NP-hard 问题。对此问题由于没有多项式最优解算法，通常采用基于贪心算法的启发式算法得到最优或次优解。

[0004] 目前以节能为目标放置虚拟机时，使用的能效模型大多是基于 CPU 一种系统资源。一些文献中指出服务器能耗与 CPU 利用率呈近似线性关系，当服务器处于空闲状态时也会消耗其处于顶峰负载时 70% 左右的电力。Beloglazov 等基于此能效模型提出了一种高能效的资源分配算法，通过虚拟机的放置和迁移有效地降低了系统能耗同时保证了任务的性能。刘志飘等基于此能效模型提出了一种基于离散粒子群方法的能量感知虚拟机放置智能优化方法。然而，在实际应用中云数据中心物理结点中的 CPU、内存、带宽、磁盘等资源的综合使用情况对系统的能效有重要的影响。Srikantaiah 等研究了系统资源对能耗、性能的影响，通过实际测量得出结论：对于非空闲的物理结点，当 CPU 利用率为 70% 且磁盘利用率为 50% 时，物理结点的能耗最低且能有效保证任务的性能。

[0005] 云数据中心虚拟机的放置是一个装箱问题，即寻找最优的虚拟机到物理结点的映射关系，使放置结果达到最优。装箱问题属于 NP-hard 问题，一般采用启发式算法，而大多启发式算法基于贪心算法，并采用一些简单规则，如次优配合、最优配合和最佳配合等，目前的虚拟机放置算法，大多是基于传统启发式算法的改进算法。使用能效模型设计虚拟机放置算法时也多采用传统启发式算法的改进算法。Beloglazov 等基于 BFD（Best Fit Decrease）算法，提出了 MBFD（Modified Best Fit Decrease）算法进行虚拟机放置。MBFD 算法首先将虚拟机按照 CPU 资源利用率进行降序排序，然后按照虚拟机请求顺序将虚拟机放置在能耗增量最小的物理结点上。Srikantaiah 等将 BFH（Best Fit Heuristics）启发

式算法进行改进，使用MBFH（Modified Best Fit Heuristics)算法进行虚拟机的放置，即将虚拟机放置在非空闲物理结点总欧式距离最小的物理结点上。基于贪心算法的传统启发式算法能优化虚拟机的放置，但是传统启发式算法通常采用单点搜索策略，容易陷入局部最优，并不能达到整体放置效果最优，仍需进一步优化。

发明内容

[0006] 针对以上现有技术中的不足，本发明的目的在于提供一种有效降低云计算数据中心的虚拟机的放置的能耗，系统资源得到合理分配的云数据中心基于多资源的高能效虚拟机放置方法。本发明的技术方案如下：一种云数据中心基于多资源的高能效虚拟机放置方法，其包括以下步骤：

[0007] 101、获取云数据中心中多资源种类数量d，云数据中心物理节点的数量n及实际利用率u_j^i，并根据实际计算环境设定能效最佳利用率$ubest_j^i$，其中u_j^i表示物理结点i上第j种资源的实际利用率，$ubest_j^i$表示物理结点i上第j种资源的能效最佳利用率，建立多资源能效模型，如式(1)所示：

[0008] $$\delta = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (u_j^i - ubest_j^i)^2} \quad (1);$$

[0009] 102、随机生成N个虚拟机请求序列，采用首次适应First Fit算法得到N种虚拟机的初始放置序列，即N个粒子，从而构成初始种群；

[0010] 103、根据步骤101中得到的多资源能效模型，$\delta = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (u_j^i - ubest_j^i)^2}$，构建粒子的适应度函数f(δ)，并根据粒子群优化算法计算出适应度f(δ)、局部最优解及全局最优解，得出粒子的新位置；

[0011] 104、根据步骤103中得到的粒子新位置，判断是否满足$\sum_h x_h^j = 1, \forall j$ (4)及

$$\sum_j r_j^{CPU} * x_h^j \le c_h^{CPU},\ \sum_j r_j^{RAM} * x_h^j \le c_h^{RAM},\ \sum_j r_j^{BW} * x_h^j \le c_h^{BW},\ \sum_j r_j^{DISK} * x_h^j \le c_h^{DISK} \quad (5)$$

[0012] 其中x_h^j表示虚拟机j是否放置在物理结点h，当虚拟机j放置在物理结点h上，x_h^j为1，否则为0；表示每个虚拟机只能放在一个物理结点上；式(5)表示多个虚拟机放置在物理结点h时，虚拟机资源不能超过物理结点h的资源总量，其中，r_j^{CPU}、r_j^{RAM}、r_j^{BW}、r_j^{DISK}分别表示虚拟机j所需的CPU、内存、带宽和磁盘的大小，c_h^{CPU}、c_h^{RAM}、c_h^{BW}、c_h^{DISK}分别表示物理结点h的CPU、内存、带宽、磁盘的容量；

[0013] 105、若满足步骤104中的式(4)和式(5)，则更新粒子的位置，迭代次数加1；若不满足式(4)和式(5)则粒子位置不变，迭代次数加1；当迭代次数≥最大迭代次数N3时，迭代结束并输出全局最优解，根据该全局最优解所对应的物理节点位置设置虚拟机，完成虚拟机的放置。

[0014] 进一步的，步骤 101 中云数据中心的多资源种类包括 CPU、内存、带宽及磁盘。

[0015] 进一步的，步骤 101 中 CPU 和磁盘的最佳能效利用率$ubest_j^i$分别为 0.7 及 0.5。

[0016] 本发明的优点及有益效果如下：

[0017] (1) 本发明方法采用多资源能效模型，能够适应云数据中心包含多种资源的虚拟机的放置问题，综合考虑不同资源对于系统能耗的影响，而不只是考虑 CPU 对于能耗的影响。

[0018] (2) 采用粒子群算法处理虚拟机放置问题，能够避免传统启发式算法进行单点搜索结果容易陷入局部最优的问题，有效降低虚拟机放置的系统能耗，系统资源得到合理分配。

[0019] (3) 该方法基于多资源的能效模型和粒子群算法，提出了高能效的虚拟机放置方法，综合考虑了能耗、性能、资源利用率等多方面的因素，保证了虚拟机放置效果。

附图说明

[0020] 图 1 是本发明优选实施例云数据中心基于多资源的高能效虚拟机放置方法流程图。

具体实施方式

[0021] 下面结合附图给出一个非限定性的实施例对本发明作进一步的阐述。

[0022] (1) 多资源能效模型建立

[0023] 数据中心物理结点的 CPU、内存、带宽、磁盘等资源的综合使用情况对物理结点的能效有重要的影响。已有文献报道通过实验研究了物理结点的能耗、性能与各种系统资源之间的关系。通过 4 个物理结点控制客户端的 k 个应用程序服务，每个物理结点连接一个测定能量的功率计和一个监控资源利用率的跟踪器，将物理结点的各种资源利用率从 10% ～ 90% 分别以 10% 的增量变化，测量不同利用率下应用程序的性能和物理结点的能耗。经过多次实验，测量结果表明，当非空闲的物理结点的 CPU 利用率为 70% 且磁盘利用率为 50% 时，能耗最小且能有效保证任务性能。使用虚拟机放置后非空闲物理结点的资源利用率最佳结合点的欧氏距离评价物理结点能效的优劣，使用所有非空闲物理结点的总欧氏距离评价系统整体能效的优劣。用公式(1) 表示：

[0024] $$\delta = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (u_j^i - ubest_j^i)^2} \qquad (1)$$

[0025] 其中 d 表示模型中资源种类的数量，资源可以是 CPU，磁盘，内存和网络带宽等，n 表示云数据中心物理节点的数量，u_j^i表示物理结点上 i 第 j 种资源的实际利用率，$ubest_j^i$表示物理结点 i 上第 j 种资源的能效最佳的利用率。数据中心所有非空闲物理结点的资源利用率与最佳结合点的欧氏距离之和，即表示当前系统与最佳状态的偏离程度。

[0026] 该能效模型根据实际测量结果确定能效最优时资源利用率的阀值，更贴近真实的云数据中心环境，对云数据中心的仿真有更大的参考价值。

[0027] 为了简化模型，本次描述中考虑两种资源 CPU 和磁盘，根据已有报道 CPU 和磁盘的最佳能效利用率分别为 0.7、0.5。

[0028] （2）虚拟机放置问题的定义

[0029] 虚拟机放置问题是一个组合优化问题，可以将其定义为多维向量装箱问题。每个物理结点的可用资源（箱子）为一个 d 维向量，每一维表示一种系统资源（CPU、内存、带宽、磁盘等）。虚拟机（物品）也是一个对应的 d 维向量。目标是使虚拟机放置后系统接近能效的最佳状态，即 δ 尽量小，如表达式（2）所示。虚拟机放置问题可以描述如下：

[0030] 目标：$\min \delta$ （2）

[0031] 约束：$x_h^j \in \{0,1\}$ （3）

[0032] $$\sum_h x_h^j = 1, \forall j \quad (4)$$

[0033] 表达式（3）中，x_h^j 表示虚拟机 j 是否放置在物理结点 h，当虚拟机 j 放置在物理结点 h 上，x_h^j 为 1，否则为 0；表达式（4）表示每个虚拟机只能放在一个物理结点上。

[0034] $$\sum_j r_j^{CPU} * x_h^j \le c_h^{CPU}, \quad \sum_j r_j^{RAM} * x_h^j \le c_h^{RAM}, \quad \sum_j r_j^{BW} * x_h^j \le c_h^{BW}, \quad \sum_j r_j^{DISK} * x_h^j \le c_h^{DISK} \quad (5)$$

[0035] 式（4）和式（5）举例

[0036] 例如，假设云计算中心拥有 4 个物理结点、4 个虚拟机；参考常用服务器的配置依次创建物理结点，物理结点配置如表 1 所示，虚拟机如表 2 所示。

[0037] 表 1 物理节点配置

[0038]

配置	CPU (MIPS)	RAM (MB)	BW (MBPS)	DISK (GB)
物理节点 1	c_1^{CPU}=600	c_1^{RAM}=2048	c_1^{BW}=1000	c_1^{DISK}=1000
物理节点 2	c_2^{CPU}=600	c_2^{RAM}=2048	c_2^{BW}=2000	c_2^{DISK}=250
物理节点 3	c_3^{CPU}=600	c_3^{RAM}=4096	c_3^{BW}=1000	c_3^{DISK}=250
物理节点 4	c_4^{CPU}=400	c_4^{RAM}=4096	c_4^{BW}=1000	c_4^{DISK}=250

[0039] 表 2 虚拟机配置

[0040]

配置	CPU (MIPS)	RAM (MB)	BW (MBPS)	DISK (GB)
虚拟机 1	r_1^{CPU}=60	r_1^{RAM}=200	r_1^{BW}=100	r_1^{DISK}=100
虚拟机 2	r_2^{CPU}=60	r_2^{RAM}=200	r_2^{BW}=30	r_2^{DISK}=200
虚拟机 3	r_3^{CPU}=60	r_3^{RAM}=400	r_3^{BW}=30	r_3^{DISK}=100
虚拟机 4	r_4^{CPU}=80	r_4^{RAM}=400	r_4^{BW}=30	r_4^{DISK}=100

[0041] 式(4)表示一个虚拟机只能放置在一个物理节点上,即:当虚拟机 1 放在在物理节点 1 时,$x_1^1=1$,而x_2^1,x_3^1,x_4^1均为 0,则有$\sum_h x_h^1=1$(h 表示 1,2,3,4),该粒子群满足式(4);

[0042] 式(5)表示多个虚拟机放置在物理结点 h 时,虚拟机资源不能超过物理结点 h 的资源总量。例如虚拟机 1,2,3,4 均放在物理节点 4 时,$x_4^1=x_4^2=x_4^3=x_4^4=1$,其余$x_h^j$为 0,,则$\sum_j r_j^{CPU}*x_4^j=60+60+60+60=240\le c_4^{CPU}(400)$,同样有$\sum_j r_j^{RAM}*x_h^j\le c_h^{RAM}$,$\sum_j r_j^{BW}*x_h^j\le c_h^{BW}$,但是$\sum_j r_j^{DISK}*x_4^j=100+200+100+100=400>c_4^{DISK}(250)$,不满足$\sum_j r_j^{DISK}*x_h^j\le c_h^{DISK}$,此时,该粒子群不满足式(5)的条件。

[0043] 表达式(5)表示多个虚拟机放置在物理结点 h 时,虚拟机资源不能超过物理结点 h 的资源总量。其中,r_j^{CPU}、r_j^{RAM}、r_j^{BW}、r_j^{DISK}分别表示虚拟机 j 所需的 CPU、内存、带宽和磁盘的大小。c_h^{CPU}、c_h^{RAM}、c_h^{BW}、c_h^{DISK}分别表示物理结点 h 的 CPU、内存、带宽、磁盘的容量。

[0044] (3)改进粒子群算法的算子和位置更新过程

[0045] 粒子群优化算法(Particle Swarm Optimizer,PSO)是由 Kenney 和 Eberhart 于 1995 年提出的,用于解决连续空间的优化问题。与其他群体演化算法相比,粒子群算法由于参数少、收敛速度快且容易实现,得到了越来越多人的关注,取得了显著成果,但是在虚拟机放置等离散组合优化问题中还有待进一步研究。由于虚拟机放置问题为离散的组合优化问题,需要重新定义粒子群算法的算子及更新过程。假设云数据中心有 m 个虚拟机,n 个物理结点。

[0046] 1)粒子位置的定义

[0047] 粒子位置向量$X_i^t=(x_{i1}^t,x_{i2}^t,...,x_{ij}^t,...,x_{im}^t)$表示虚拟机放置的第 1 种可行方案,其中$x_{ij}^t$表示在第 t 代更新中,虚拟机 j 所在的物理结点的序号。假设对于虚拟机放置的第 1 种可行方案(x_{i1}^t,x_{i2}^t,x_{i3}^t)分别为(1,2,1),表示虚拟机 1、2、3 分别放置在 1、2、1 号物理结点上。对粒子位置进行更新时,粒子的位置向量x_i^t转换成一个 0、1 位置矩阵XX_i^t:

[0048]
$$XX_i^t=\begin{bmatrix} s_{11}^t & s_{12}^t & \cdots & s_{1h}^t & \cdots & s_{1n}^t \\ s_{21}^t & s_{22}^t & \cdots & s_{2h}^t & \cdots & s_{2n}^t \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{j1}^t & s_{j2}^t & \cdots & s_{jh}^t & \cdots & s_{jn}^t \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{m1}^t & s_{m2}^t & \cdots & s_{mh}^t & \cdots & s_{mn}^t \end{bmatrix} \quad (6)$$

[0049] 公式(6)中,s_{jh}^t表示在第 t 代更新中虚拟机 j 是否放在物理结点 h 上,如果虚拟机 j 放在物理结点 h 上,则$s_{jh}^t=1$,否则$s_{jh}^t=0$。s_{jh}^t的值根据x_i^t的第 j 维的值而定。例如,若x_i^t为(1,2,1),第一维x_{i1}^t为 1 表示虚拟机放在了第 1 个物理结点上,则XX_i^t的第一行中s_{11}^t为 1,同理s_{22}^t为 1,s_{31}^t为 1。由于一个虚拟机只能放在一个物理结点上,所以$\sum_{h=1}^{n} s_{jh}^t=1,\forall j\in\{1,2,...,m\}$。

[0050] 2）位置的更新过程

[0051] 本文根据每个粒子的个体最优解和所有粒子的全局最优解对粒子位置进行更新，$Pbest_l^t=(pbest_{l1}^t,pbest_{l2}^t,...,pbest_{lj}^t,...,pbest_{lm}^t)$表示第 l 个粒子经过 t 次迭代后的个体最优解，其中$pbest_{lj}^t$表示粒子 l 个体最优解第 j 维的位置坐标，$Gbest^t=(gbest_1^t,gbest_2^t,...,gbest_j^t,...,gbest_m^t)$表示所有粒子经过 t 次迭代后的全局最优解，$gbest_j^t$表示所有粒子全局最优解的第 j 维的位置坐标。$Pbest_l^t$、Gbestt 在粒子更新的过程中也将转换成相应的位置矩阵，公式（7）～（10）中的$pbest_{jh}^t$、$gbest_{jh}^t$即是对应位置矩阵的第 j 行、h 列的值。

[0052] 本文假设在随机状态下，第 t+1 次位置更新时，虚拟机 j 放或者不放在物理结点 h 上的概率为 0.5，即$P(s_{jh}^{t+1}=1)=P(s_{jh}^{t+1}=0)=0.5$。使用 p_p、p_g 分别表示个体最优解和全局最优解的信任度，即发现粒子最终最优解的概率，以此判断粒子的位置。为了描述简便，本文引入两个参数 k_1、k_2。由贝叶斯公式得：

[0053]
$$P(s_{jh}^{t+1}=1\mid pbest_{jh}^t=1,gbest_{jh}^t=1)=\frac{p_p*p_g}{p_p*p_g+(1-p_p)(1-p_g)}=k_1 \quad (7)$$

[0054]
$$P(s_{jh}^{t+1}=0\mid pbest_{jh}^t=1,gbest_{jh}^t=1)=\frac{(1-p_p)*(1-p_g)}{p_p*p_g+(1-p_p)(1-p_g)}=1-k_1 \quad (8)$$

[0055]
$$P(s_{jh}^{t+1}=1\mid pbest_{jh}^t=1,gbest_{jh}^t=0)=\frac{p_p*(1-p_g)}{p_p*(1-p_g)+p_p*(1-p_g)}=k_2 \quad (9)$$

[0056]
$$P(s_{jh}^{t+1}=0\mid pbest_{jh}^t=1,gbest_{jh}^t=0)=\frac{p_g*(1-p_p)}{p_p*(1-p_g)+p_g*(1-p_p)}=1-k_2 \quad (10)$$

[0057] 公式（7）～（10）表示粒子位置坐标根据个体最优解和全局最优解取 0 或者 1 的概率。由于个体最优解和全局最优解是迭代过程产生的历史最优解，找到最优解的概率应该大于平均值，即 $0.5<p_p<1$，$0.5<p_g<1$，为了使粒子进行分散搜索，避免粒子陷入局部最优，应令 $p_g<p_p$，即 $0.5<p_g<p_p<1$。本文取 $p_p=0.7$，$p_g=0.8$，对应的 $k_1=0.9$，$k_2=0.7$。

[0058] 位置更新过程的伪代码如下：

[0059]

```
r=rand();
```

[0060]

if($pbest_{jh}^{t}$ =1 and $gbest_{jh}^{t}$ =1) then

if(r< k_1) s_{jh}^{t+1} =1 else s_{jh}^{t+1} =0;

else if($pbest_{jh}^{t}$ =0 and $gbest_{jh}^{t}$ =0) then

if(r<1- k_1) s_{jh}^{t+1} =0 else s_{jh}^{t+1} =1;

else if($pbest_{jh}^{t}$ =1 and $gbest_{jh}^{t}$ =0) then

if(r< k_2) s_{jh}^{t+1} =1 else s_{jh}^{t+1} =0;

else

if(r<1- k_2) s_{jh}^{t+1} =1 else s_{jh}^{t+1} =0;

[0061] (4)算法流程描述

[0062] 1)初始化种群

[0063] 随机生成N个虚拟机请求序列,对每一个请求序列,按照首次适应(First Fit)算法,将虚拟机放置在第一个满足资源的物理结点上,得到N种放置方案,即N个粒子,从而构成初始种群,其时间复杂度为o(nlogn)。

[0064] 2)根据适应度函数,计算初始种群的适应度,得到粒子个体最优解和全局最优解。

[0065] 3)根据4.2.1(2)计算粒子的新位置。

[0066] 4)判断新位置是否满足要求,然后将迭代次数加1。依次判断每个粒子的每个虚拟机的新位置是否满足公式(4)、(5)的约束条件,如果所有虚拟机都满足约束条件,则粒子更新为新的位置,否则粒子保持原来的值。判断公式(4)、(5)的约束条件是否满足的方法如下:对于第j个虚拟机,①如果$\sum_{h=1}^{n} s_{jh}^{t} > 1$,即虚拟机j放在多个物理结点上,则记录位置为1的物理结点的序号,然后设置$\forall h, s_{jh}^{t} = 0$。按照物理结点序号递增的顺序,依次取所记录的物理结点,将虚拟机j放在满足(5)的第一个物理结点上,并将其位置置1,接着判断粒子的其他虚拟机。如果记录的物理结点都不满足(5),则该虚拟机新位置不满足条件,粒子不更新。②如果$\sum_{h=1}^{n} s_{jh}^{t} = 1$,即虚拟机放在某个物理结点上,判断是否满足约束(5),如果满足,则虚拟机新位置符合条件,粒子更新为新位置,否则粒子不更新。③如果$\sum_{h=1}^{n} s_{jh}^{t} = 0$,即更新后虚拟机没有分配给任何主机,粒子新位置不满足条件,粒子不更新。

[0067] 5)判断是否为最大迭代次数,若是,则结束,输出全局最优解,若否,则转到(2)。

[0068] 本文方法在二进制离散粒子群算法的基础上进行改进,不再通过速度转换得到粒子的位置,而是通过概率直接确定粒子新位置,搜索更直接。算法时间复杂度为o(mn)。与传统启发式算法相比,本文方法通过粒子比较进化得到整体放置效果最优的放置结果,能有效避免陷入局部最优解。

[0069] 以上这些实施例应理解为仅用于说明本发明而不用于限制本发明的保护范围。在

阅读了本发明的记载的内容之后，技术人员可以对本发明作各种改动或修改，这些等效变化和修饰同样落入本发明方法权利要求所限定的范围。

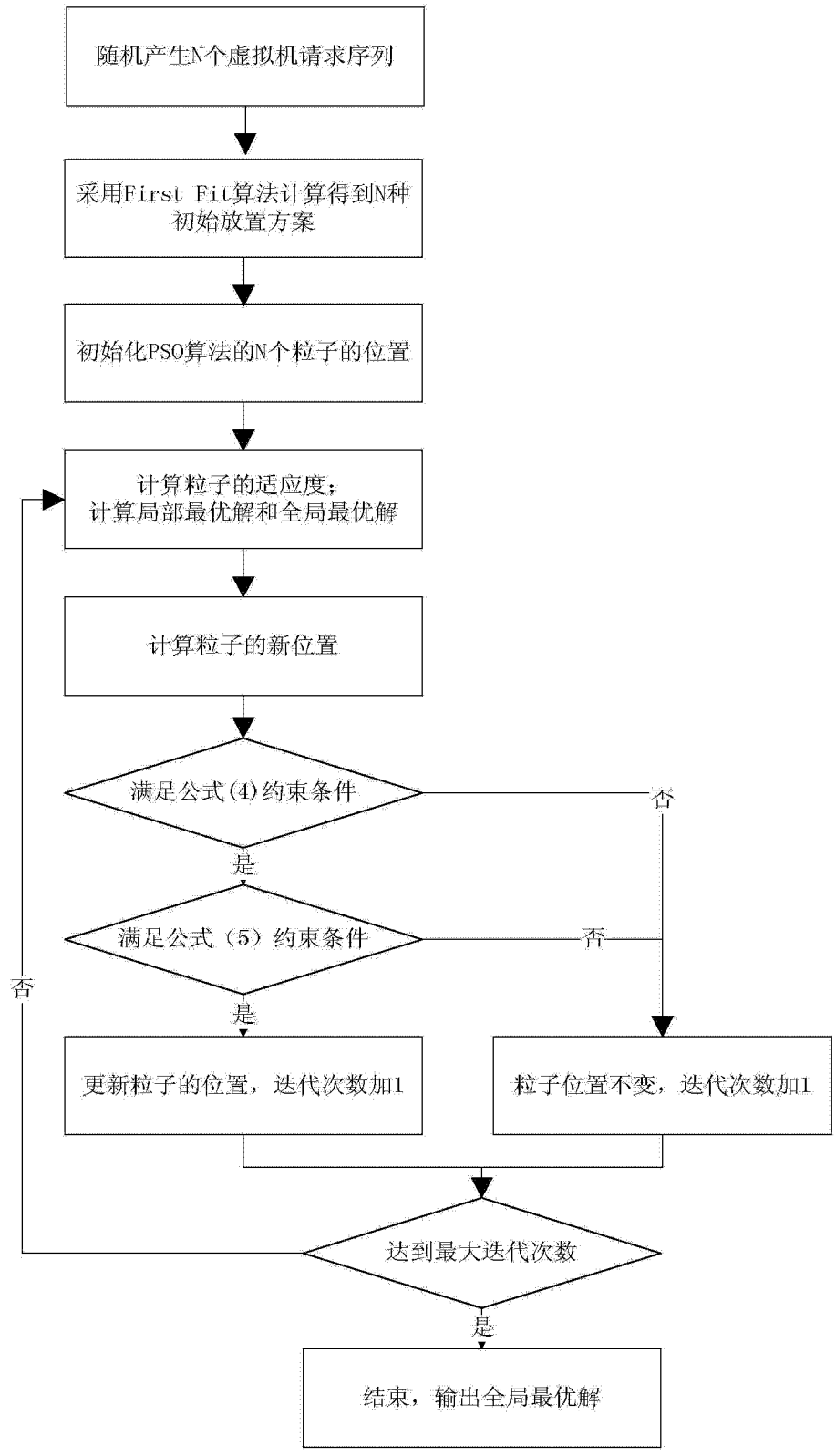

随机产生N个虚拟机请求序列
采用First Fit算法计算得到N种初始放置方案
初始化PSO算法的N个粒子的位置
计算粒子的适应度；计算局部最优解和全局最优解
计算粒子的新位置
满足公式(4)约束条件
否
是
满足公式（5）约束条件
否
是
否
更新粒子的位置，迭代次数加1
粒子位置不变，迭代次数加1
达到最大迭代次数
是
结束，输出全局最优解

图 1