

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2022 年 5 月 19 日 (19.05.2022)

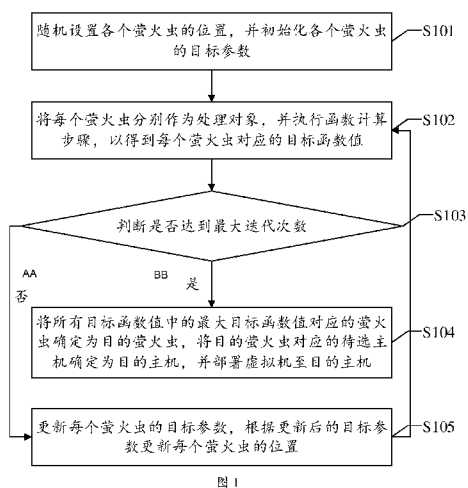


(10) 国际公布号
WO 2022/100306 A1

- (51) 国际专利分类号:
G06F 9/455 (2006.01) **G06N 3/00** (2006.01)
- (21) 国际申请号: PCT/CN2021/121216
- (22) 国际申请日: 2021 年 9 月 28 日 (28.09.2021)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202011261567.6 2020 年 11 月 12 日 (12.11.2020) CN
- (71) 申请人: 苏州浪潮智能科技有限公司
(**INSPUR SUZHOU INTELLIGENT TECHNOLOGY CO., LTD.**) [CN/CN]; 中国江苏省苏州市吴中区吴中经济开发区郭巷街道官浦路 1 号 9 幢, Jiangsu 215000 (CN)。
- (72) 发明人: 沈新新(**SHEN, Xinxin**); 中国山东省济南市高新区浪潮路 1036 号, Shandong 250101 (CN)。
- (74) 代理人: 北京三聚阳光知识产权代理有限公司 (**SUNSHINE INTELLECTUAL PROPERTY INTERNATIONAL CO., LTD.**); 中国北京市海淀区海淀南路甲 21 号中关村知识产权大厦 A 座 5 层 503, Beijing 100080 (CN)。
- (81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

(54) **Title:** VIRTUAL MACHINE DEPLOYMENT METHOD AND APPARATUS, DEVICE, AND READABLE STORAGE MEDIUM

(54) 发明名称: 一种虚拟机部署方法、装置、设备及可读存储介质



- 图 1
- S101 Randomly arrange locations of respective fireflies, and initialize target parameters of the respective fireflies
- S102 Take each firefly as an object to be processed, and execute a function calculation step to obtain a target function value corresponding to each firefly
- S103 Determine whether a maximum iteration count is reached
- S104 Determine a firefly corresponding to the maximum target function value in all target function values to be a destination firefly, determine a host to be selected corresponding to the destination firefly to be a destination host, and deploy virtual machines to the destination host
- S105 Update the target parameter of each firefly, and update the location of each firefly according to the updated target parameter
- AA No
- BB Yes

(57) **Abstract:** A virtual machine deployment method and apparatus, a device, and a readable storage medium. The method uses a firefly algorithm to determine deployment locations of virtual machines, and takes an average performance score of all hosts to be selected as a target function. Locations of respective fireflies are locations of the respective hosts. An iterative optimization process of the firefly algorithm involves finding a host capable of maximizing operation performance of all hosts in a cloud platform after deployment of virtual machines is completed. Since the target function is the average performance score of all hosts after deployment of the virtual machines, selection of a host corresponding to the maximum target function value enables average performance of all hosts to be maximized after the virtual machines have been deployed to a destination host, thereby maximizing operation performance of all hosts in the cloud platform while scheduling the virtual machines.

(57) **摘要:** 一种虚拟机部署方法、装置、设备及可读存储介质。该方法利用萤火虫算法确定虚拟机的部署位置, 并且以所有待选主机的平均性能大小作为目标函数, 各个萤火虫的位置即各个待选主机的位置, 因此萤火虫算法的迭代寻优过程即为: 寻找虚拟机部署完成后, 能使云平台中所有主机的运行性能最大化的主机。由于目标函数为虚拟机部署后, 所有待选主机的平均性能大小, 因此选择最大目标函数值对应的待选主机, 就能使虚拟机部署到目的主机后, 所有待选主机的平均性能最大化, 从而实现了在调度虚拟机的同时, 使云平台中所有主机的运行性能最大化。

ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US,
UZ, VC, VN, WS, ZA, ZM, ZW。

- (84) 指定国 (除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

根据细则4.17的声明:

- 发明人资格 (细则4.17(iv))

本国际公布:

- 包括国际检索报告 (条约第21条(3))。

一种虚拟机部署方法、装置、设备及可读存储介质

本申请要求在 2020 年 11 月 12 日提交中国专利局、申请号为 202011261567.6、发明名称为“一种虚拟机部署方法、装置、设备及可读存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机技术领域，特别涉及一种虚拟机部署方法、装置、设备及可读存储介质。

背景技术

云平台中的虚拟机可以在不同主机上进行调度。一般地，可以选择负载较小的主机部署新虚拟机，或者将负载较大的主机上的虚拟机迁移至负载较小的主机，从而使各主机的负载尽可能均衡。如此调度虚拟机，虽然可以避免个别主机的负载过大或过小，但没有考虑到整个云平台的对外服务能力，难以保障云平台中所有主机的运行性能。

因此，如何在调度虚拟机的同时，使云平台中所有主机的运行性能最大化，是本领域技术人员需要解决的问题。

发明内容

有鉴于此，本申请的目的在于提供一种虚拟机部署方法、装置、设备及可读存储介质，以在调度虚拟机的同时，使云平台中所有主机的运行性能最大化。其具体方案如下：

第一方面，本申请提供了一种虚拟机部署方法，包括：

随机设置各个萤火虫的位置，并初始化各个萤火虫的目标参数；所述各个萤火虫的位置即各个待选主机的位置；

将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值；所述函数计算步骤包括：根据所述处理对象的位置计算目标函数，所述目标函数能够表示：假定虚拟机部署至所述处理对象后，所有待选主机的平均性能大小；

判断是否达到最大迭代次数；

若达到最大迭代次数, 则将所有目标函数值中的最大目标函数值对应的萤火虫确定为目的萤火虫, 将所述目的萤火虫对应的待选主机确定为目的主机, 并部署所述虚拟机至所述目的主机;

若未达到最大迭代次数, 则更新每个萤火虫的目标参数, 根据更新后的所述目标参数更新每个萤火虫的位置, 并执行将每个萤火虫分别作为处理对象, 并执行函数计算步骤, 以得到每个萤火虫对应的目标函数值的步骤。

优选地, 所述目标函数的计算公式为:

$$F = K_1 * \frac{f_p}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_2 * \frac{f_{load}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_3 * \frac{f_{CPU}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}}$$

其中, F 为所述目标函数; f_p 为假定所述虚拟机部署至所述处理对象后, 所有待选主机的平均电源消耗; f_{CPU} 为假定所述虚拟机部署至所述处理对象后, 所有待选主机的平均 CPU 利用率; f_{load} 为假定所述虚拟机部署至所述处理对象后, 所有待选主机的平均资源均衡度; K_1 、 K_2 、 K_3 均为预设权重值。

优选地, 所述平均资源均衡度是假定所述虚拟机部署至所述处理对象后, 根据所有待选主机的平均 CPU 利用率、平均内存利用率、平均磁盘利用率和平均带宽利用率计算获得的; 所述平均资源均衡度的计算公式为:

$$\left\{ \begin{array}{l} f_{load} = \frac{4}{\left(\frac{u_{cpu}}{u}\right)^2 + \left(\frac{u_{mem}}{u}\right)^2 + \left(\frac{u_{hw}}{u}\right)^2 + \left(\frac{u_{disk}}{u}\right)^2} \\ u = \frac{u_{cpu} + u_{mem} + u_{hw} + u_{disk}}{4} \end{array} \right.$$

其中, f_{load} 为所述平均资源均衡度, u_{cpu} 为所述平均 CPU 利用率, u_{mem} 为所述平均内存利用率, u_{hw} 为所述平均带宽利用率, u_{disk} 为所述平均磁盘利用率。

优选地, 所述目标参数包括荧光素和步长, 所述更新每个萤火虫的目标参数, 包括: 利用第一公式更新每个萤火虫的荧光素, 所述第一公式为:

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma F_i(t+1)$$

利用第二公式更新每个萤火虫的步长, 所述第二公式为:

$$\left\{ \begin{array}{l} s_i(t+1) = s_{min} + ((s_{max} - s_{min})l_i(t))^c \\ c = 1 + \frac{t+1}{t_{max}} - \cos\left(\frac{t+1}{t_{max}}\right) \end{array} \right.$$

其中, $l_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的荧光素, ρ ($0 < \rho < 1$) 为萤火虫 i 的荧光素挥发速度, $l_i(t)$ 为萤火虫 i 在第 t 次迭代时的荧光素, γ 为萤火虫 i 的荧光素更新率, $F_i(t+1)$

为萤火虫*i*在第*t*+1次迭代时的目标函数值； $s_i(t+1)$ 为萤火虫*i*在第*t*+1次迭代时的步长， $s_{\min}$ 为预设最小步长， $s_{\max}$ 为预设最大步长，*t*+1为当前迭代次数， $t_{\max}$ 为所述最大迭代次数。

优选地，所述根据更新后的所述目标参数更新每个萤火虫的位置，包括：

利用第三公式更新每个萤火虫的位置，所述第三公式为：

$$x_i(t+1) = x_i(t) + s_i(t+1) \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right)$$

其中， $x_i(t+1)$ 为萤火虫*i*在第*t*+1次迭代时的位置， $x_i(t)$ 为萤火虫*i*在第*t*次迭代时的位置， $s_i(t+1)$ 为萤火虫*i*在第*t*+1次迭代时的步长， $x_j(t)$ 为萤火虫*j*在第*t*次迭代时的位置， $\|x_j(t) - x_i(t)\|$ 为萤火虫*i*与萤火虫*j*位置的欧式距离；

其中，萤火虫*j*为基于轮盘赌法则，在萤火虫*i*第*t*次迭代时的决策域内选择的邻居萤火虫，且萤火虫*j*的荧光素大于萤火虫*i*的荧光素。

优选地，所述目标参数还包括决策域，利用第四公式更新每个萤火虫的决策域，所述第四公式为：

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\}$$

其中， $r_d^i(t+1)$ 为萤火虫*i*在第*t*+1次迭代时的决策域， r_s 为预设感知域， $r_d^i(t)$ 为萤火虫*i*在第*t*次迭代时的决策域， β 为动态决策域更新率， n_i 为邻域集合阈值， $N_i(t)$ 为萤火虫*i*在第*t*次迭代时的邻域集合。

优选地，所述平均电源消耗、所述平均 CPU 利用率和所述平均资源均衡度均经过归一化处理，归一化公式为：

$$\begin{cases} f_p = 1 - \frac{f_{p_o} - f_{p_min}}{f_{p_max} - f_{p_min}} \\ f_{cpu} = \frac{f_{cpu_o} - f_{cpu_min}}{f_{cpu_max} - f_{cpu_min}} \\ f_{load} = \frac{f_{load_o} - f_{load_min}}{f_{load_max} - f_{load_min}} \end{cases}$$

其中， f_p 为归一化后的所述平均电源消耗， f_{p_o} 为归一化前的所述平均电源消耗， f_{p_min} 为最小平均电源消耗， f_{p_max} 为最大平均电源消耗； f_{cpu} 为归一化后的所述平均 CPU 利用率， f_{cpu_o} 为归一化前的所述平均 CPU 利用率， f_{cpu_min} 为最小平均 CPU 利用率， f_{cpu_max} 为最大平均 CPU 利用率； f_{load} 为归一化后的所述平均资源均衡度， f_{load_o} 为归一化前的所述平均资源均衡度， f_{load_min} 为最小平均资源均衡度， f_{load_max} 为最大平均资源均衡度。

第二方面，本申请提供了一种虚拟机部署装置，包括：

初始化模块，用于随机设置各个萤火虫的位置，并初始化各个萤火虫的目标参数；所述各个萤火虫的位置即各个待选主机的位置；

函数计算模块，用于将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值；所述函数计算步骤包括：根据所述处理对象的位置计算目标函数，所述目标函数能够表示：假定虚拟机部署至所述处理对象后，所有待选主机的平均性能大小；

判断模块，用于判断是否达到最大迭代次数；

部署模块，用于若达到最大迭代次数，则将所有目标函数值中的最大目标函数值对应的萤火虫确定为目的萤火虫，将所述目的萤火虫对应的待选主机确定为目的主机，并部署所述虚拟机至所述目的主机；

迭代模块，用于若未达到最大迭代次数，则更新每个萤火虫的目标参数，根据更新后的所述目标参数更新每个萤火虫的位置，并执行将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值的步骤。

第三方面，本申请提供了一种虚拟机部署设备，包括：

存储器，用于存储计算机程序；

处理器，用于执行所述计算机程序，以实现前述公开的虚拟机部署方法。

第四方面，本申请提供了一种可读存储介质，用于保存计算机程序，其中，所述计算机程序被处理器执行时实现前述公开的虚拟机部署方法。

通过以上方案可知，本申请提供了一种虚拟机部署方法，包括：随机设置各个萤火虫的位置，并初始化各个萤火虫的目标参数；所述各个萤火虫的位置即各个待选主机的位置；将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值；所述函数计算步骤包括：根据所述处理对象的位置计算目标函数，所述目标函数能够表示：假定虚拟机部署至所述处理对象后，所有待选主机的平均性能大小；判断是否达到最大迭代次数；若达到最大迭代次数，则将所有目标函数值中的最大目标函数值对应的萤火虫确定为目的萤火虫，将所述目的萤火虫对应的待选主机确定为目的主机，并部署所述虚拟机至所述目的主机；若未达到最大迭代次数，则更新每个萤火虫的目标参数，根据更新后的所述目标参数更新每个萤火虫的位置，并执行将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值的步骤。

可见，本申请利用萤火虫算法确定虚拟机的部署位置，并且以所有待选主机的平均性能大小作为目标函数，各个萤火虫的位置即各个待选主机的位置，因此萤火虫算法的迭代寻优过程即为：寻找虚拟机部署完成后，能使云平台中所有主机的运行性能最大化的主机。由于目标函数为虚拟机部署后，所有待选主机的平均性能大小，因此选择最大目标函数值对应的待选主机，就能使虚拟机部署到目的主机后，所有待选主机的平均性能最大化，从而实现了在调度虚拟机的同时，使云平台中所有主机的运行性能最大化。

相应地，本申请提供一种虚拟机部署装置、设备及可读存储介质，也同样具有上述技术效果。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据提供的附图获得其他的附图。

图 1 为本申请公开的一种虚拟机部署方法流程图；

图 2 为本申请公开的一种利用萤火虫算法求解目标函数的方法流程图；

图 3 为本申请公开的一种虚拟机部署装置示意图；

图 4 为本申请公开的一种虚拟机部署设备示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

在介绍本申请之前，先就相关背景技术进行如下介绍。

一般在下述三种场景中需要进行虚拟机调度。一：用户创建新虚拟机，选择将该虚拟机部署到哪台物理主机上。二：一个物理主机上的虚拟机过多，负载过重，需要将该物理主机上的部分虚拟机迁移至其他物理主机上；或者一个物理主机上的虚拟机过少，负载较低，为了实现负载均衡，需要将其他物理主机上的虚拟机迁移至当前物理主机。三：物理

主机故障，需要疏散该物理主机上的虚拟机至其他物理主机上。不合理的调度策略会引起虚拟机的多次跨物理主机迁移，严重影响整个云平台的稳定性。并且，使用云平台的每个用户所申请的资源及应用不尽相同，每个物理主机上的负载也不尽相同，也导致云平台的资源使用具有不稳定性。

目前可以使用蚁群算法寻找部署虚拟机的目的主机，但由于蚁群算法存在求解速度慢且所得解的质量难以保障的问题，导致计算量大，收敛速度慢，易陷入局部最优，不能在调度虚拟机的同时，使云平台中所有主机的运行性能最大化。为此，本申请提供了一种虚拟机部署方案，能够在调度虚拟机的同时，使云平台中所有主机的运行性能最大化。

参见图 1 所示，本申请实施例公开了一种虚拟机部署方法，包括：

S101、随机设置各个萤火虫的位置，并初始化各个萤火虫的目标参数。

其中，各个萤火虫的位置即各个待选主机的位置。萤火虫的位置和待选主机的个数相等，但萤火虫的个数与待选主机的个数可以相等，也可以不等。若萤火虫的个数大于待选主机的个数，那么一个萤火虫的位置可能聚集有多个萤火虫。

S102、将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值。

其中，函数计算步骤包括：根据处理对象的位置计算目标函数，目标函数能够表示：假定虚拟机部署至处理对象后，所有待选主机的平均性能大小。

在一种具体实施方式中，目标函数的计算公式为：

$$F = K_1 * \frac{f_p}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_2 * \frac{f_{load}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_3 * \frac{f_{CPU}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}}$$

其中， F 为目标函数； f_p 为假定虚拟机部署至处理对象后，所有待选主机的平均电源消耗； f_{CPU} 为假定虚拟机部署至处理对象后，所有待选主机的平均 CPU 利用率； f_{load} 为假定虚拟机部署至处理对象后，所有待选主机的平均资源均衡度； K_1 、 K_2 、 K_3 均为预设权重值。预设权重值越大，表明当前云平台侧重关注其对应的参数（电源消耗、CPU 利用率或资源均衡度）。

例如：针对萤火虫 A，认为其对应待选主机 A，那么假定虚拟机部署至待选主机 A，此时基于虚拟机的电源消耗、CPU 利用率、带宽利用率、内存利用率等相关参数调整待选主机 A 的电源消耗、CPU 利用率、资源均衡度。然后计算某一时刻所有待选主机（包括部

署有虚拟机的待选主机 A) 的平均电源消耗、平均 CPU 利用率以及平均资源均衡度。F 越大，表明所有待选主机的平均性能越好。

在一种具体实施方式中，平均资源均衡度是假定虚拟机部署至处理对象后，根据所有待选主机某一时刻的平均 CPU 利用率、平均内存利用率、平均磁盘利用率和平均带宽利用率计算获得的；平均资源均衡度的计算公式为：

$$\left\{ \begin{array}{l} f_{load} = \frac{4}{\left(\frac{u_{cpu}}{u}\right)^2 + \left(\frac{u_{mem}}{u}\right)^2 + \left(\frac{u_{hw}}{u}\right)^2 + \left(\frac{u_{disk}}{u}\right)^2} \\ u = \frac{u_{cpu} + u_{mem} + u_{hw} + u_{disk}}{4} \end{array} \right.$$

其中， f_{load} 为平均资源均衡度， u_{cpu} 为平均 CPU 利用率， u_{mem} 为平均内存利用率， u_{hw} 为平均带宽利用率， u_{disk} 为平均磁盘利用率。

在一种具体实施方式中，平均电源消耗、平均 CPU 利用率和平均资源均衡度均经过归一化处理，归一化公式为：

$$\left\{ \begin{array}{l} f_p = 1 - \frac{f_{p_o} - f_{p_min}}{f_{p_max} - f_{p_min}} \\ f_{cpu} = \frac{f_{cpu_o} - f_{cpu_min}}{f_{cpu_max} - f_{cpu_min}} \\ f_{load} = \frac{f_{load_o} - f_{load_min}}{f_{load_max} - f_{load_min}} \end{array} \right.$$

其中， f_p 为归一化后的平均电源消耗， f_{p_o} 为归一化前的平均电源消耗， f_{p_min} 为最小平均电源消耗， f_{p_max} 为最大平均电源消耗； f_{cpu} 为归一化后的平均 CPU 利用率， f_{cpu_o} 为归一化前的平均 CPU 利用率， f_{cpu_min} 为最小平均 CPU 利用率， f_{cpu_max} 为最大平均 CPU 利用率； f_{load} 为归一化后的平均资源均衡度， f_{load_o} 为归一化前的平均资源均衡度， f_{load_min} 为最小平均资源均衡度， f_{load_max} 为最大平均资源均衡度。归一化后的平均电源消耗、平均 CPU 利用率和平均资源均衡度取值在 0~1 之间。

S103、判断是否达到最大迭代次数；若达到最大迭代次数，则执行 S104；若未达到最大迭代次数，则执行 S105。

S104、将所有目标函数值中的最大目标函数值对应的萤火虫确定为目的萤火虫，将目的萤火虫对应的待选主机确定为目的主机，并部署虚拟机至目的主机。

S105、更新每个萤火虫的目标参数，根据更新后的目标参数更新每个萤火虫的位置，并执行 S102。

在一种具体实施方式中, 目标参数包括荧光素和步长, 更新每个萤火虫的目标参数, 包括:

利用第一公式更新每个萤火虫的荧光素, 第一公式为:

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma F_i(t+1)$$

利用第二公式更新每个萤火虫的步长, 第二公式为:

$$\begin{cases} s_i(t+1) = s_{\min} + ((s_{\max} - s_{\min})l_i(t))^c \\ c = 1 + \frac{t+1}{t_{\max}} - \cos(\frac{t+1}{t_{\max}}) \end{cases}$$

其中, $l_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的荧光素, 也就是当前更新后的荧光素, ρ ($0 < \rho < 1$) 为萤火虫 i 的荧光素挥发速度, $l_i(t)$ 为萤火虫 i 在第 t 次迭代时的荧光素, γ 为萤火虫 i 的荧光素更新率, $F_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的目标函数值; $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长, 也就是当前更新后的步长; $s_{\min}$ 为预设最小步长, 取值为接近于 0 的值; $s_{\max}$ 为预设最大步长, 取值为接近于 1 的值; $t+1$ 为当前迭代次数, $t_{\max}$ 为最大迭代次数。

需要说明的是, 每个萤火虫的步长会随着迭代次数的递增而变化, 从而可保障萤火虫的步长保持在合理范围内。

在一种具体实施方式中, 根据更新后的目标参数更新每个萤火虫的位置, 包括:

利用第三公式更新每个萤火虫的位置, 第三公式为:

$$x_i(t+1) = x_i(t) + s_i(t+1) \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right)$$

其中, $x_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的位置, $x_i(t)$ 为萤火虫 i 在第 t 次迭代时的位置, $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长, $x_j(t)$ 萤火虫 j 在第 t 次迭代时的位置, $\|x_j(t) - x_i(t)\|$ 为萤火虫 i 与萤火虫 j 位置的欧式距离;

其中, 萤火虫 j 为基于轮盘赌法则, 在萤火虫 i 第 t 次迭代时的决策域内选择的邻居萤火虫, 且萤火虫 j 的荧光素大于萤火虫 i 的荧光素。

在一种具体实施方式中, 目标参数还包括决策域, 利用第四公式更新每个萤火虫的决策域, 第四公式为:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\}$$

其中, $r_d^i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的决策域, r_s 为预设感知域, $r_d^i(t)$ 为萤火虫 i 在第 t 次迭代时的决策域, β 为动态决策域更新率, n_i 为邻域集合阈值, $N_i(t)$ 为萤火虫 i 在第 t 次迭代时的邻域集合, 即萤火虫 i 在第 t 次迭代时的决策域内的邻居萤火虫的个数。

可见, 本申请实施例利用萤火虫算法确定虚拟机的部署位置, 并且以所有待选主机的平均性能大小作为目标函数, 各个萤火虫的位置即各个待选主机的位置, 因此萤火虫算法的迭代寻优过程即为: 寻找虚拟机部署完成后, 能使云平台中所有主机的运行性能最大化的主机。由于目标函数为虚拟机部署后, 所有待选主机的平均性能大小, 因此选择最大目标函数值对应的待选主机, 就能使虚拟机部署到目的主机后, 所有待选主机的平均性能最大化, 从而实现了在调度虚拟机的同时, 使云平台中所有主机的运行性能最大化。

本申请实施例公开了一种虚拟机部署方案, 包括: 定义目标函数、求解目标函数和部署虚拟机。

1、假定虚拟机部署在各个待选主机上, 定义目标函数 F :

$$F = K_1 * \frac{f_p}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_2 * \frac{f_{load}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_3 * \frac{f_{CPU}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}}$$

其中, f_p 为所有待选主机的平均电源消耗; f_{CPU} 为所有待选主机的平均 CPU 利用率; f_{load} 为所有待选主机的平均资源均衡度; K_1 、 K_2 、 K_3 均为预设权重值。

其中,

$$\left\{ \begin{array}{l} f_{load} = \frac{4}{\left(\frac{u_{cpu}}{u}\right)^2 + \left(\frac{u_{mem}}{u}\right)^2 + \left(\frac{u_{hw}}{u}\right)^2 + \left(\frac{u_{disk}}{u}\right)^2} \\ u = \frac{u_{cpu} + u_{mem} + u_{hw} + u_{disk}}{4} \end{array} \right.$$

其中, f_{load} 为平均资源均衡度, u_{cpu} 为平均 CPU 利用率, u_{mem} 为平均内存利用率, u_{hw} 为平均带宽利用率, u_{disk} 为平均磁盘利用率。

其中, f_p 、 f_{cpu} 、 f_{load} 均为归一化后的值, 归一化公式为:

$$\left\{ \begin{array}{l} f_p = 1 - \frac{f_{p_o} - f_{p_min}}{f_{p_max} - f_{p_min}} \\ f_{cpu} = \frac{f_{cpu_o} - f_{cpu_min}}{f_{cpu_max} - f_{cpu_min}} \\ f_{load} = \frac{f_{load_o} - f_{load_min}}{f_{load_max} - f_{load_min}} \end{array} \right.$$

其中, f_p 为归一化后的平均电源消耗, f_{p_o} 为归一化前的平均电源消耗, f_{p_min} 为最小平均电源消耗, f_{p_max} 为最大平均电源消耗; f_{cpu} 为归一化后的平均 CPU 利用率, f_{cpu_o} 为归一化前的平均 CPU 利用率, f_{cpu_min} 为最小平均 CPU 利用率, f_{cpu_max} 为最大平均 CPU

利用率； f_{load} 为归一化后的平均资源均衡度， f_{load_o} 为归一化前的平均资源均衡度， f_{load_min} 为最小平均资源均衡度， f_{load_max} 为最大平均资源均衡度。

2、参见图 2 所示，利用萤火虫算法求解目标函数：

(1) 参数初始化：

初始化各个萤火虫的位置；各个萤火虫的位置即各个待选主机的位置。

初始化各个萤火虫的目标参数，如：荧光素、步长、决策域。

设定各个预设值，如：荧光素挥发速度 ρ ($0 < \rho < 1$)，荧光素更新率 γ ，最小步长 s_{min} 、最大步长 s_{max} 、最大迭代次数 t_{max} 、感知域 r_s 、邻域集合阈值 n_i 。

(2) 分别假定虚拟机部署在各个待选主机上，以基于萤火虫的当前位置计算各个萤火虫对应的目标函数值。目标函数值越大，则所有待选主机的平均性能越好。

(3) 更新每个萤火虫的荧光素：

萤火虫的荧光素与其所在位置有直接关系。荧光素越大，则其所位置的适应度值越高。荧光素的大小还与上一次迭代中荧光素的大小及其挥发速度，以及目标函数值有关，其更新公式为：

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma F_i(t+1)$$

其中， $l_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的荧光素，也就是当前更新后的荧光素， ρ ($0 < \rho < 1$) 为萤火虫 i 的荧光素挥发速度， $l_i(t)$ 为萤火虫 i 在第 t 次迭代时的荧光素， γ 为萤火虫 i 的荧光素更新率， $F_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的目标函数值。

(4) 每个萤火虫的步长：

在算法迭代的初期，步长保持较大值。为避免算法过早的成熟，陷入局部最优；在算法迭代的后期，逐渐减小步长，以避免在算法迭代后期因为步长太大而跳过最优解。步长更新公式为：

$$\begin{cases} s_i(t+1) = s_{min} + ((s_{max} - s_{min})l_i(t))^c \\ c = 1 + \frac{t+1}{t_{max}} - \cos(\frac{t+1}{t_{max}}) \end{cases}$$

其中， $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长，也就是当前更新后的步长； s_{min} 为预设最小步长，取值为接近于 0 的值； s_{max} 为预设最大步长，取值为接近于 1 的值； $l_i(t)$ 为萤火虫 i 在第 t 次迭代时的荧光素； $t+1$ 为当前迭代次数， t_{max} 为最大迭代次数。

(5) 更新每个萤火虫的位置：

每个萤火虫按照步长移动位置之前，需要先找到一个符合条件的邻居萤火虫。找邻居萤火虫的条件为：该邻居萤火虫的位置在当前萤火虫的决策域范围内，且邻居萤火虫的荧光素大于当前萤火虫的荧光素。将满足上述条件的邻居萤火虫组合为邻域集合，并按照下述公式计算邻域集合中每个邻居萤火虫的概率：

$$p_{ij} = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}$$

其中， $j \in N_i(t)$ ， $N_i(t) = \{j; d_{i,j}(t) < r_d^i(t); l_j(t) < l_i(t)\}$ ， $N_i(t)$ 是在第 t 次迭代中，任一个萤火虫 i 的邻域集合， $d_{i,j}(t) = \|x_j(t) - x_i(t)\|$ 是在第 t 次迭代中萤火虫 i 与萤火虫 j 的欧式距离。

基于每个邻居萤火虫的概率，用轮盘赌法则选出邻居萤火虫 j ，然后根据下式更新萤火虫 i 的空间位置：

$$x_i(t+1) = x_i(t) + s_i(t+1) \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right)$$

其中， $x_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的位置， $x_i(t)$ 为萤火虫 i 在第 t 次迭代时的位置， $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长， $x_j(t)$ 萤火虫 j 在第 t 次迭代时的位置， $\|x_j(t) - x_i(t)\|$ 为萤火虫 i 与萤火虫 j 位置的欧式距离。

(6) 更新每个萤火虫的决策域：

萤火虫的位置更新完成之后，会进一步更新其决策域，更新公式为：

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\}$$

其中， $r_d^i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的决策域， r_s 为预设感知域， $r_d^i(t)$ 为萤火虫 i 在第 t 次迭代时的决策域， β 为动态决策域更新率， n_i 为邻域集合阈值， $N_i(t)$ 为萤火虫 i 在第 t 次迭代时的邻域集合，即萤火虫 i 在第 t 次迭代时的决策域内的邻居萤火虫的个数。

(7) 判断此刻的迭代次数：

若当前迭代次数大于最大迭代次数 $t_{\max}$ ，则结束迭代，并用最新的萤火虫位置计算目标函数值，选择最大目标函数值以及相应的萤火虫位置输出；若当前迭代次数不大于最大迭代次数 $t_{\max}$ ，返回步骤(2)进行下次迭代，直至满足迭代次数的约束条件。

3、部署虚拟机：最大目标函数值对应的萤火虫位置即虚拟机的部署位置，将虚拟机部署至此位置，可使所有待选主机的运行性能最大。

可见，本实施例考虑了由CPU利用率、内存利用率、带宽利用率和磁盘利用率而得的资源均衡度，并结合CPU利用率和电源消耗进行归一化，定义了能够表示所有待选主机的

平均性能大小的目标函数。使用改进的萤火虫算法求解目标函数，考虑了荧光素大小以及步长大小对于迭代过程的影响，使算法具有较强的局部搜索能力，能够在一个较小的区域内找到该区域的最优解，操作方便，实现简单。并且萤火虫算法的参数较少，参数对算法的影响也小。本实施例使用的萤火虫算法可以避免个体在峰值附近发生震荡现象，通过迭代寻优之后输出最佳物理主机位置作为该虚拟机的目的物理主机，使虚拟机部署后，云平台保持 CPU 高利用率，耗能较低，资源均衡，以提升系统整体性能。

下面对本申请实施例提供的一种虚拟机部署装置进行介绍，下文描述的一种虚拟机部署装置与上文描述的一种虚拟机部署方法可以相互参照。

参见图 3 所示，本申请实施例公开了一种虚拟机部署装置，包括：

初始化模块 301，用于随机设置各个萤火虫的位置，并初始化各个萤火虫的目标参数；各个萤火虫的位置即各个待选主机的位置；

函数计算模块 302，用于将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值；函数计算步骤包括：根据处理对象的位置计算目标函数，目标函数能够表示：假定虚拟机部署至处理对象后，所有待选主机的平均性能大小；

判断模块 303，用于判断是否达到最大迭代次数；

部署模块 304，用于若达到最大迭代次数，则将所有目标函数值中的最大目标函数值对应的萤火虫确定为目的萤火虫，将目的萤火虫对应的待选主机确定为目的主机，并部署虚拟机至目的主机；

迭代模块 305，用于若未达到最大迭代次数，则更新每个萤火虫的目标参数，根据更新后的目标参数更新每个萤火虫的位置，并执行将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值的步骤。

在一种具体实施方式中，目标函数的计算公式为：

$$F = K_1 * \frac{f_p}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_2 * \frac{f_{load}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_3 * \frac{f_{CPU}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}}$$

其中， F 为目标函数； f_p 为假定虚拟机部署至处理对象后，所有待选主机的平均电源消耗； f_{CPU} 为假定虚拟机部署至处理对象后，所有待选主机的平均 CPU 利用率； f_{load} 为假定虚拟机部署至处理对象后，所有待选主机的平均资源均衡度； K_1 、 K_2 、 K_3 均为预设权重值。

在一种具体实施方式中, 平均资源均衡度是假定虚拟机部署至处理对象后, 根据所有待选主机的平均 CPU 利用率、平均内存利用率、平均磁盘利用率和平均带宽利用率计算获得的; 平均资源均衡度的计算公式为:

$$\begin{cases} f_{load} = \frac{4}{\left(\frac{u_{cpu}}{u}\right)^2 + \left(\frac{u_{mem}}{u}\right)^2 + \left(\frac{u_{hw}}{u}\right)^2 + \left(\frac{u_{disk}}{u}\right)^2} \\ u = \frac{u_{cpu} + u_{mem} + u_{hw} + u_{disk}}{4} \end{cases}$$

其中, f_{load} 为平均资源均衡度, u_{cpu} 为平均 CPU 利用率, u_{mem} 为平均内存利用率, u_{hw} 为平均带宽利用率, u_{disk} 为平均磁盘利用率。

在一种具体实施方式中, 迭代模块具体用于:

目标参数包括荧光素和步长, 利用第一公式更新每个萤火虫的荧光素, 第一公式为:

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma F_i(t+1)$$

利用第二公式更新每个萤火虫的步长, 第二公式为:

$$\begin{cases} s_i(t+1) = s_{\min} + ((s_{\max} - s_{\min})l_i(t))^c \\ c = 1 + \frac{t+1}{t_{\max}} - \cos\left(\frac{t+1}{t_{\max}}\right) \end{cases}$$

其中, $l_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的荧光素, ρ ($0 < \rho < 1$) 为萤火虫 i 的荧光素挥发速度, $l_i(t)$ 为萤火虫 i 在第 t 次迭代时的荧光素, γ 为萤火虫 i 的荧光素更新率, $F_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的目标函数值; $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长, $s_{\min}$ 为预设最小步长, $s_{\max}$ 为预设最大步长, $t+1$ 为当前迭代次数, $t_{\max}$ 为最大迭代次数。

在一种具体实施方式中, 迭代模块具体用于:

利用第三公式更新每个萤火虫的位置, 第三公式为:

$$x_i(t+1) = x_i(t) + s_i(t+1) \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right)$$

其中, $x_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的位置, $x_i(t)$ 为萤火虫 i 在第 t 次迭代时的位置, $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长, $x_j(t)$ 为萤火虫 j 在第 t 次迭代时的位置, $\|x_j(t) - x_i(t)\|$ 为萤火虫 i 与萤火虫 j 位置的欧式距离;

其中, 萤火虫 j 为基于轮盘赌法则, 在萤火虫 i 第 t 次迭代时的决策域内选择的邻居萤火虫, 且萤火虫 j 的荧光素大于萤火虫 i 的荧光素。

在一种具体实施方式中, 目标参数还包括决策域, 利用第四公式更新每个萤火虫的决策域, 第四公式为:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\}$$

其中, $r_d^i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的决策域, r_s 为预设感知域, $r_d^i(t)$ 为萤火虫 i 在第 t 次迭代时的决策域, β 为动态决策域更新率, n_i 为邻域集合阈值, $N_i(t)$ 为萤火虫 i 在第 t 次迭代时的邻域集合。

在一种具体实施方式中, 平均电源消耗、平均 CPU 利用率和平均资源均衡度均经过归一化处理, 归一化公式为:

$$\begin{cases} f_p = 1 - \frac{f_{p_o} - f_{p_min}}{f_{p_max} - f_{p_min}} \\ f_{cpu} = \frac{f_{cpu_o} - f_{cpu_min}}{f_{cpu_max} - f_{cpu_min}} \\ f_{load} = \frac{f_{load_o} - f_{load_min}}{f_{load_max} - f_{load_min}} \end{cases}$$

其中, f_p 为归一化后的平均电源消耗, f_{p_o} 为归一化前的平均电源消耗, f_{p_min} 为最小平均电源消耗, f_{p_max} 为最大平均电源消耗; f_{cpu} 为归一化后的平均 CPU 利用率, f_{cpu_o} 为归一化前的平均 CPU 利用率, f_{cpu_min} 为最小平均 CPU 利用率, f_{cpu_max} 为最大平均 CPU 利用率; f_{load} 为归一化后的平均资源均衡度, f_{load_o} 为归一化前的平均资源均衡度, f_{load_min} 为最小平均资源均衡度, f_{load_max} 为最大平均资源均衡度。

其中, 关于本实施例中各个模块、单元更加具体的工作过程可以参考前述实施例中公开的相应内容, 在此不再进行赘述。

可见, 本实施例提供了一种虚拟机部署装置, 该装置能在调度虚拟机的同时, 使云平台中所有主机的运行性能最大化。

下面对本申请实施例提供的一种虚拟机部署设备进行介绍, 下文描述的一种虚拟机部署设备与上文描述的一种虚拟机部署方法及装置可以相互参照。

参见图 4 所示, 本申请实施例公开了一种虚拟机部署设备, 包括:

存储器 401, 用于保存计算机程序;

处理器 402, 用于执行所述计算机程序, 以实现上述任意实施例公开的方法。

下面对本申请实施例提供的一种可读存储介质进行介绍, 下文描述的一种可读存储介质与上文描述的一种虚拟机部署方法、装置及设备可以相互参照。

一种可读存储介质，用于保存计算机程序，其中，所述计算机程序被处理器执行时实现前述实施例公开的虚拟机部署方法。关于该方法的具体步骤可以参考前述实施例中公开的相应内容，在此不再进行赘述。

本申请涉及的“第一”、“第二”、“第三”、“第四”等(如果存在)是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法或设备固有的其它步骤或单元。

需要说明的是，在本申请中涉及“第一”、“第二”等的描述仅用于描述目的，而不能理解为指示或暗示其相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。另外，各个实施例之间的技术方案可以相互结合，但是必须是以本领域普通技术人员能够实现为基础，当技术方案的结合出现相互矛盾或无法实现时应当认为这种技术方案的结合不存在，也不在本申请要求的保护范围之内。

本说明书中各个实施例采用递进的方式描述，每个实施例重点说明的都是与其它实施例的不同之处，各个实施例之间相同或相似部分互相参见即可。

结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块，或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的可读存储介质中。

本文中应用了具体个例对本申请的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本申请的方法及其核心思想；同时，对于本领域的一般技术人员，依据本申请的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本申请的限制。

权 利 要 求 书

1、一种虚拟机部署方法，其特征在于，包括：

随机设置各个萤火虫的位置，并初始化各个萤火虫的目标参数；所述各个萤火虫的位置即各个待选主机的位置；

将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值；所述函数计算步骤包括：根据所述处理对象的位置计算目标函数，所述目标函数能够表示：假定虚拟机部署至所述处理对象后，所有待选主机的平均性能大小；

判断是否达到最大迭代次数；

若达到最大迭代次数，则将所有目标函数值中的最大目标函数值对应的萤火虫确定为目的萤火虫，将所述目的萤火虫对应的待选主机确定为目的主机，并部署所述虚拟机至所述目的主机；

若未达到最大迭代次数，则更新每个萤火虫的目标参数，根据更新后的所述目标参数更新每个萤火虫的位置，并执行将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值的步骤。

2、根据权利要求 1 所述的虚拟机部署方法，其特征在于，所述目标函数的计算公式为：

$$F = K_1 * \frac{f_p}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_2 * \frac{f_{load}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}} + K_3 * \frac{f_{CPU}}{\sqrt{f_p^2 + f_{load}^2 + f_{CPU}^2}}$$

其中， F 为所述目标函数； f_p 为假定所述虚拟机部署至所述处理对象后，所有待选主机的平均电源消耗； f_{CPU} 为假定所述虚拟机部署至所述处理对象后，所有待选主机的平均 CPU 利用率； f_{load} 为假定所述虚拟机部署至所述处理对象后，所有待选主机的平均资源均衡度； K_1 、 K_2 、 K_3 均为预设权重值。

3、根据权利要求 2 所述的虚拟机部署方法，其特征在于，所述平均资源均衡度是假定所述虚拟机部署至所述处理对象后，根据所有待选主机的平均 CPU 利用率、平均内存利用率、平均磁盘利用率和平均带宽利用率计算获得的；所述平均资源均衡度的计算公式为：

$$\left\{ \begin{array}{l} f_{load} = \frac{4}{\left(\frac{u_{cpu}}{u}\right)^2 + \left(\frac{u_{mem}}{u}\right)^2 + \left(\frac{u_{hw}}{u}\right)^2 + \left(\frac{u_{disk}}{u}\right)^2} \\ u = \frac{u_{cpu} + u_{mem} + u_{hw} + u_{disk}}{4} \end{array} \right.$$

其中, f_{load} 为所述平均资源均衡度, u_{cpu} 为所述平均 CPU 利用率, u_{mem} 为所述平均内存利用率, u_{hw} 为所述平均带宽利用率, u_{disk} 为所述平均磁盘利用率。

4、根据权利要求 1 所述的虚拟机部署方法, 其特征在于, 所述目标参数包括荧光素和步长, 所述更新每个萤火虫的目标参数, 包括:

利用第一公式更新每个萤火虫的荧光素, 所述第一公式为:

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma F_i(t+1)$$

利用第二公式更新每个萤火虫的步长, 所述第二公式为:

$$\begin{cases} s_i(t+1) = s_{\min} + ((s_{\max} - s_{\min})l_i(t))^c \\ c = 1 + \frac{t+1}{t_{\max}} - \cos(\frac{t+1}{t_{\max}}) \end{cases}$$

其中, $l_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的荧光素, $\rho(0 < \rho < 1)$ 为萤火虫 i 的荧光素挥发速度, $l_i(t)$ 为萤火虫 i 在第 t 次迭代时的荧光素, γ 为萤火虫 i 的荧光素更新率, $F_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的目标函数值; $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长, $s_{\min}$ 为预设最小步长, $s_{\max}$ 为预设最大步长, $t+1$ 为当前迭代次数, $t_{\max}$ 为所述最大迭代次数。

5、根据权利要求 4 所述的虚拟机部署方法, 其特征在于, 所述根据更新后的所述目标参数更新每个萤火虫的位置, 包括:

利用第三公式更新每个萤火虫的位置, 所述第三公式为:

$$x_i(t+1) = x_i(t) + s_i(t+1) \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right)$$

其中, $x_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的位置, $x_i(t)$ 为萤火虫 i 在第 t 次迭代时的位置, $s_i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的步长, $x_j(t)$ 为萤火虫 j 在第 t 次迭代时的位置, $\|x_j(t) - x_i(t)\|$ 为萤火虫 i 与萤火虫 j 位置的欧式距离;

其中, 萤火虫 j 为基于轮盘赌法则, 在萤火虫 i 第 t 次迭代时的决策域内选择的邻居萤火虫, 且萤火虫 j 的荧光素大于萤火虫 i 的荧光素。

6、根据权利要求 5 所述的虚拟机部署方法, 其特征在于, 所述目标参数还包括决策域, 利用第四公式更新每个萤火虫的决策域, 所述第四公式为:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\}$$

其中, $r_d^i(t+1)$ 为萤火虫 i 在第 $t+1$ 次迭代时的决策域, r_s 为预设感知域, $r_d^i(t)$ 为萤火虫 i 在第 t 次迭代时的决策域, β 为动态决策域更新率, n_i 为邻域集合阈值, $N_i(t)$ 为萤火虫 i 在第 t 次迭代时的邻域集合。

7、根据权利要求 3 所述的虚拟机部署方法，其特征在于，所述平均电源消耗、所述平均 CPU 利用率和所述平均资源均衡度均经过归一化处理，归一化公式为：

$$\begin{cases} f_p = 1 - \frac{f_{p_o} - f_{p_{\min}}}{f_{p_{\max}} - f_{p_{\min}}} \\ f_{cpu} = \frac{f_{cpu_o} - f_{cpu_{\min}}}{f_{cpu_{\max}} - f_{cpu_{\min}}} \\ f_{load} = \frac{f_{load_o} - f_{load_{\min}}}{f_{load_{\max}} - f_{load_{\min}}} \end{cases}$$

其中， f_p 为归一化后的所述平均电源消耗， f_{p_o} 为归一化前的所述平均电源消耗， $f_{p_{\min}}$ 为最小平均电源消耗， $f_{p_{\max}}$ 为最大平均电源消耗； f_{cpu} 为归一化后的所述平均 CPU 利用率， f_{cpu_o} 为归一化前的所述平均 CPU 利用率， $f_{cpu_{\min}}$ 为最小平均 CPU 利用率， $f_{cpu_{\max}}$ 为最大平均 CPU 利用率； f_{load} 为归一化后的所述平均资源均衡度， f_{load_o} 为归一化前的所述平均资源均衡度， $f_{load_{\min}}$ 为最小平均资源均衡度， $f_{load_{\max}}$ 为最大平均资源均衡度。

8、一种虚拟机部署装置，其特征在于，包括：

初始化模块，用于随机设置各个萤火虫的位置，并初始化各个萤火虫的目标参数；所述各个萤火虫的位置即各个待选主机的位置；

函数计算模块，用于将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值；所述函数计算步骤包括：根据所述处理对象的位置计算目标函数，所述目标函数能够表示：假定虚拟机部署至所述处理对象后，所有待选主机的平均性能大小；

判断模块，用于判断是否达到最大迭代次数；

部署模块，用于若达到最大迭代次数，则将所有目标函数值中的最大目标函数值对应的萤火虫确定为目的萤火虫，将所述目的萤火虫对应的待选主机确定为目的主机，并部署所述虚拟机至所述目的主机；

迭代模块，用于若未达到最大迭代次数，则更新每个萤火虫的目标参数，根据更新后的所述目标参数更新每个萤火虫的位置，并执行将每个萤火虫分别作为处理对象，并执行函数计算步骤，以得到每个萤火虫对应的目标函数值的步骤。

9、一种虚拟机部署设备，其特征在于，包括：

存储器，用于存储计算机程序；

处理器，用于执行所述计算机程序，以实现如权利要求 1 至 7 任一项所述的虚拟机部署方法。

10、一种可读存储介质，其特征在于，用于保存计算机程序，其中，所述计算机程序被处理器执行时实现如权利要求 1 至 7 任一项所述的虚拟机部署方法。

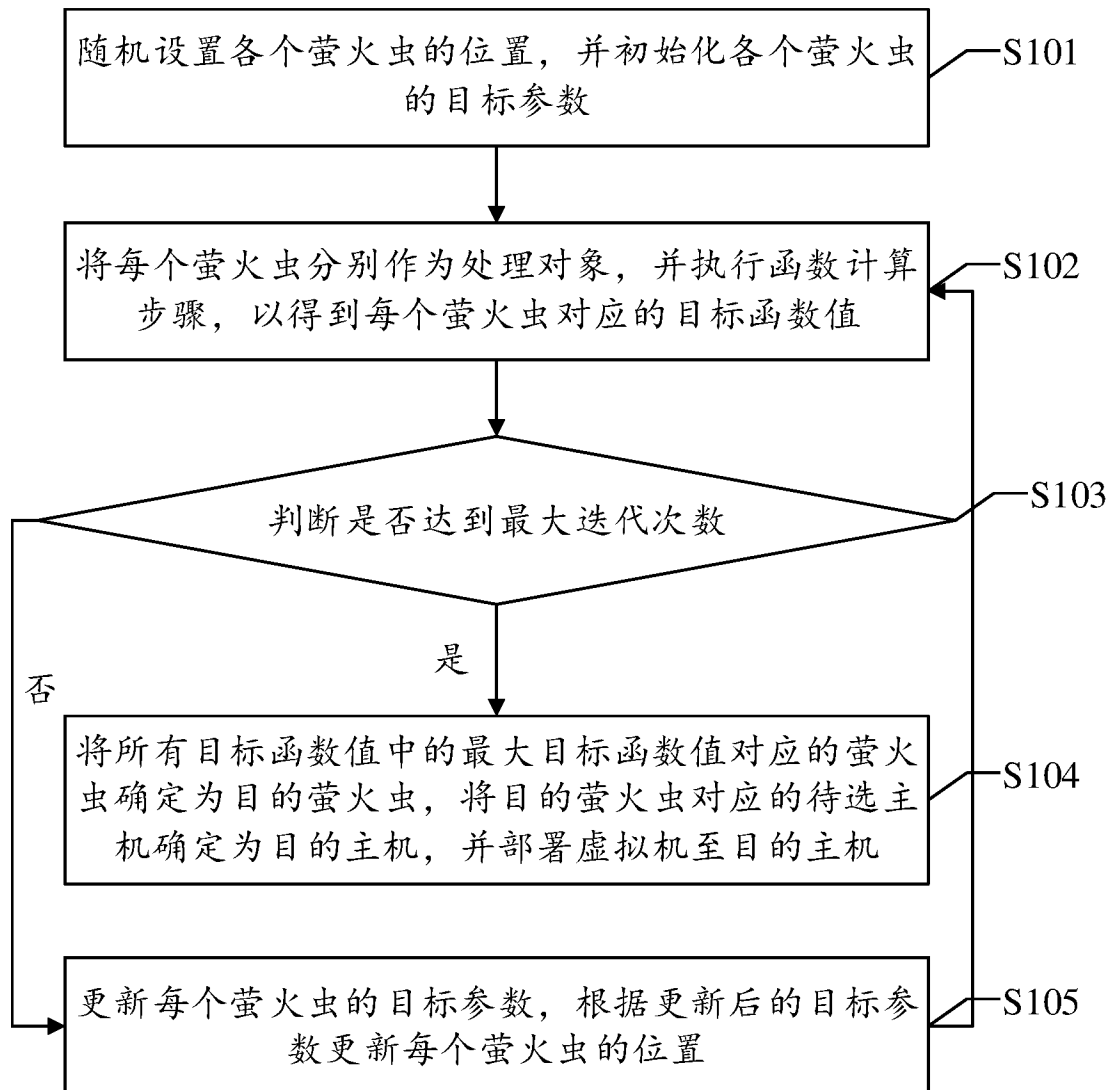


图 1

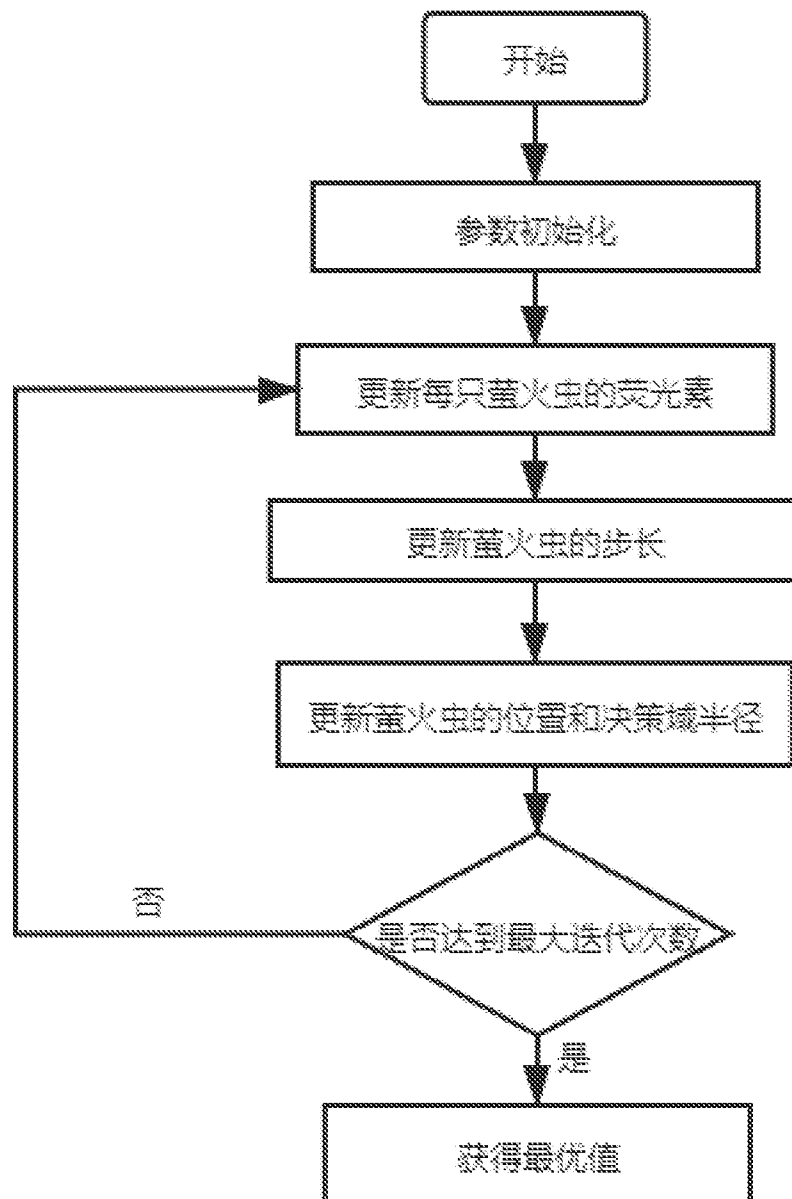


图 2

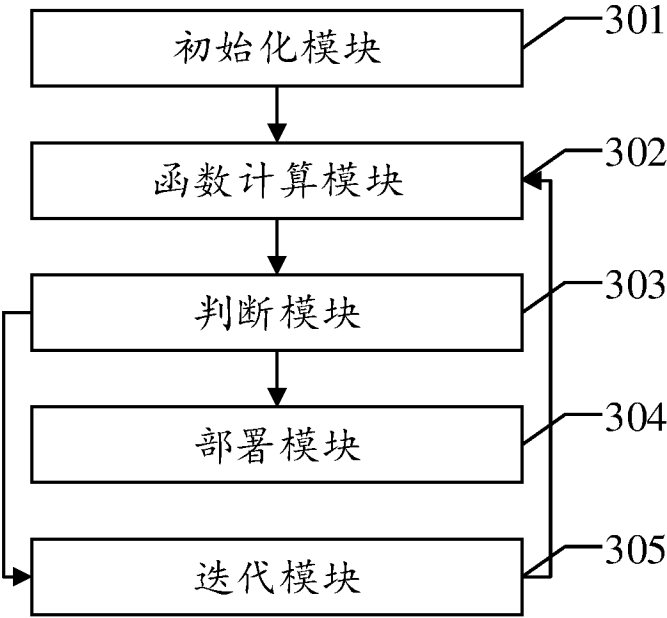


图 3

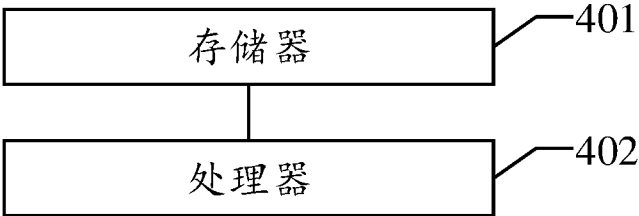


图 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2021/121216

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/455(2006.01)i; G06N 3/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F; G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, WPI, EPODOC, CNKI, GOOGLE, ISI: 虚拟机, 虚机, 部署, 分配, 配置, 调度, 迁移, 萤火虫, 优化, 平均, 总, 性能, 功耗, 电源, 利用率, 资源, 均衡, 步长; virtual machine, placement, optimization, glowworm, total, average, energy, power, source, consumption, balance, step

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 112463291 A (SUZHOU INSPUR INTELLIGENT TECHNOLOGY CO., LTD.) 09 March 2021 (2021-03-09) claims 1-10	1-10
X	龚婷婷 等 (GONG, Tingting et al.). "基于萤火虫算法的虚拟机部署能效优化策略 (Virtual Machine Placement Energy-efficient Optimization based on Glowworm Swarm Optimization Algorithm)" <i>计算机工程与设计 (Computer Engineering and Design)</i> , Vol. 40, No. 7, 31 July 2019 (2019-07-31), pages 1897-1900, and figure 1	1, 8-10
X	倪志伟 等 (NI, Zhiwei et al.). "面向数据中心虚拟机部署的智能优化策略 (Intelligent Optimization Strategy for Virtual Machine Placement in Data Center)" <i>模式识别与人工智能 (Pattern Recognition and Artificial Intelligence)</i> , Vol. 28, No. 4, 28 April 2015 (2015-04-28), pp. 307-310	1, 8-10
A	CN 110362379 A (NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY) 22 October 2019 (2019-10-22) entire document	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

15 December 2021

Date of mailing of the international search report

30 December 2021

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088, China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2021/121216**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017337086 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 23 November 2017 (2017-11-23) entire document	1-10
A	CN 110120892 A (SHANDONG TECHNOLOGY AND BUSINESS UNIVERSITY) 13 August 2019 (2019-08-13) entire document	1-10
A	CN 103366362 A (KUNMING UNIVERSITY OF SCIENCE AND TECHNOLOGY) 23 October 2013 (2013-10-23) entire document	1-10

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2021/121216

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	112463291	A	09 March 2021	None			
CN	110362379	A	22 October 2019	None			
US	2017337086	A1	23 November 2017	EP	3247089	A1	22 November 2017
				EP	3247089	B1	03 April 2019
				US	10289451	B2	14 May 2019
				CN	104639639	A	20 May 2015
				WO	2016127547	A1	18 August 2016
CN	110120892	A	13 August 2019	None			
CN	103366362	A	23 October 2013	CN	103366362	B	20 January 2016

国际检索报告

国际申请号

PCT/CN2021/121216

A. 主题的分类 G06F 9/455 (2006.01) i; G06N 3/00 (2006.01) i 按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类																							
B. 检索领域 检索的最低限度文献 (标明分类系统和分类号) G06F; G06N 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用)) CNPAT, WPI, EPDOC, CNKI, GOOGLE, ISI: 虚拟机, 虚机, 部署, 分配, 配置, 调度, 迁移, 萤火虫, 优化, 平均, 总, 性能, 功耗, 电源, 利用率, 资源, 均衡, 步长; virtual machine, placement, optimization, glowworm, total, average, energy, power, source, consumption, balance, step																							
C. 相关文件 <table border="1"> <thead> <tr> <th>类 型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>PX</td> <td>CN 112463291 A (苏州浪潮智能科技有限公司) 2021年3月9日 (2021 - 03 - 09) 权利要求1-10</td> <td>1-10</td> </tr> <tr> <td>X</td> <td>龚婷婷 等. “基于萤火虫算法的虚拟机部署能效优化策略” 计算机工程与设计, 第40卷, 第7期, 2019年7月31日 (2019 - 07 - 31), 第1897-1900页、图1</td> <td>1, 8-10</td> </tr> <tr> <td>X</td> <td>倪志伟 等. “面向数据中心虚拟机部署的智能优化策略” 模式识别与人工智能, 第28卷, 第4期, 2015年4月28日 (2015 - 04 - 28), 第307-310页</td> <td>1, 8-10</td> </tr> <tr> <td>A</td> <td>CN 110362379 A (南京理工大学) 2019年10月22日 (2019 - 10 - 22) 全文</td> <td>1-10</td> </tr> <tr> <td>A</td> <td>US 2017337086 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 2017年11月23日 (2017 - 11 - 23) 全文</td> <td>1-10</td> </tr> <tr> <td>A</td> <td>CN 110120892 A (山东工商学院) 2019年8月13日 (2019 - 08 - 13) 全文</td> <td>1-10</td> </tr> </tbody> </table>			类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求	PX	CN 112463291 A (苏州浪潮智能科技有限公司) 2021年3月9日 (2021 - 03 - 09) 权利要求1-10	1-10	X	龚婷婷 等. “基于萤火虫算法的虚拟机部署能效优化策略” 计算机工程与设计, 第40卷, 第7期, 2019年7月31日 (2019 - 07 - 31), 第1897-1900页、图1	1, 8-10	X	倪志伟 等. “面向数据中心虚拟机部署的智能优化策略” 模式识别与人工智能, 第28卷, 第4期, 2015年4月28日 (2015 - 04 - 28), 第307-310页	1, 8-10	A	CN 110362379 A (南京理工大学) 2019年10月22日 (2019 - 10 - 22) 全文	1-10	A	US 2017337086 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 2017年11月23日 (2017 - 11 - 23) 全文	1-10	A	CN 110120892 A (山东工商学院) 2019年8月13日 (2019 - 08 - 13) 全文	1-10
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求																					
PX	CN 112463291 A (苏州浪潮智能科技有限公司) 2021年3月9日 (2021 - 03 - 09) 权利要求1-10	1-10																					
X	龚婷婷 等. “基于萤火虫算法的虚拟机部署能效优化策略” 计算机工程与设计, 第40卷, 第7期, 2019年7月31日 (2019 - 07 - 31), 第1897-1900页、图1	1, 8-10																					
X	倪志伟 等. “面向数据中心虚拟机部署的智能优化策略” 模式识别与人工智能, 第28卷, 第4期, 2015年4月28日 (2015 - 04 - 28), 第307-310页	1, 8-10																					
A	CN 110362379 A (南京理工大学) 2019年10月22日 (2019 - 10 - 22) 全文	1-10																					
A	US 2017337086 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 2017年11月23日 (2017 - 11 - 23) 全文	1-10																					
A	CN 110120892 A (山东工商学院) 2019年8月13日 (2019 - 08 - 13) 全文	1-10																					
☒ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。																							
<table border="0"> <tr> <td style="vertical-align: top;"> * 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 </td> <td style="vertical-align: top;"> “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件 </td> </tr> </table>			* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件																			
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件																						
国际检索实际完成的日期		国际检索报告邮寄日期																					
2021年12月15日		2021年12月30日																					
ISA/CN的名称和邮寄地址		授权官员																					
中国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451		张琳琳 电话号码 86-(10)-53961404																					

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	CN 103366362 A（昆明理工大学）2013年10月23日（2013 - 10 - 23） 全文	1-10

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2021/121216

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	112463291	A	2021年3月9日	无			
CN	110362379	A	2019年10月22日	无			
US	2017337086	A1	2017年11月23日	EP	3247089	A1	2017年11月22日
				EP	3247089	B1	2019年4月3日
				US	10289451	B2	2019年5月14日
				CN	104639639	A	2015年5月20日
				WO	2016127547	A1	2016年8月18日
CN	110120892	A	2019年8月13日	无			
CN	103366362	A	2013年10月23日	CN	103366362	B	2016年1月20日