US 20250265104A1

(19) United States
(12) Patent Application Publication (10) Pub. No.: US 2025/0265104 A1
Shen
(43) Pub. Date: Aug. 21, 2025

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

(71) Applicant: **INSPUR SUZHOU INTELLIGENT TECHNOLOGY CO., LTD.**, Jiangsu (CN)

(72) Inventor: **Xinxin Shen**, Shandong (CN)

(73) Assignee: **INSPUR SUZHOU INTELLIGENT TECHNOLOGY CO., LTD.**, Jiangsu (CN)

Publication Classification

(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.
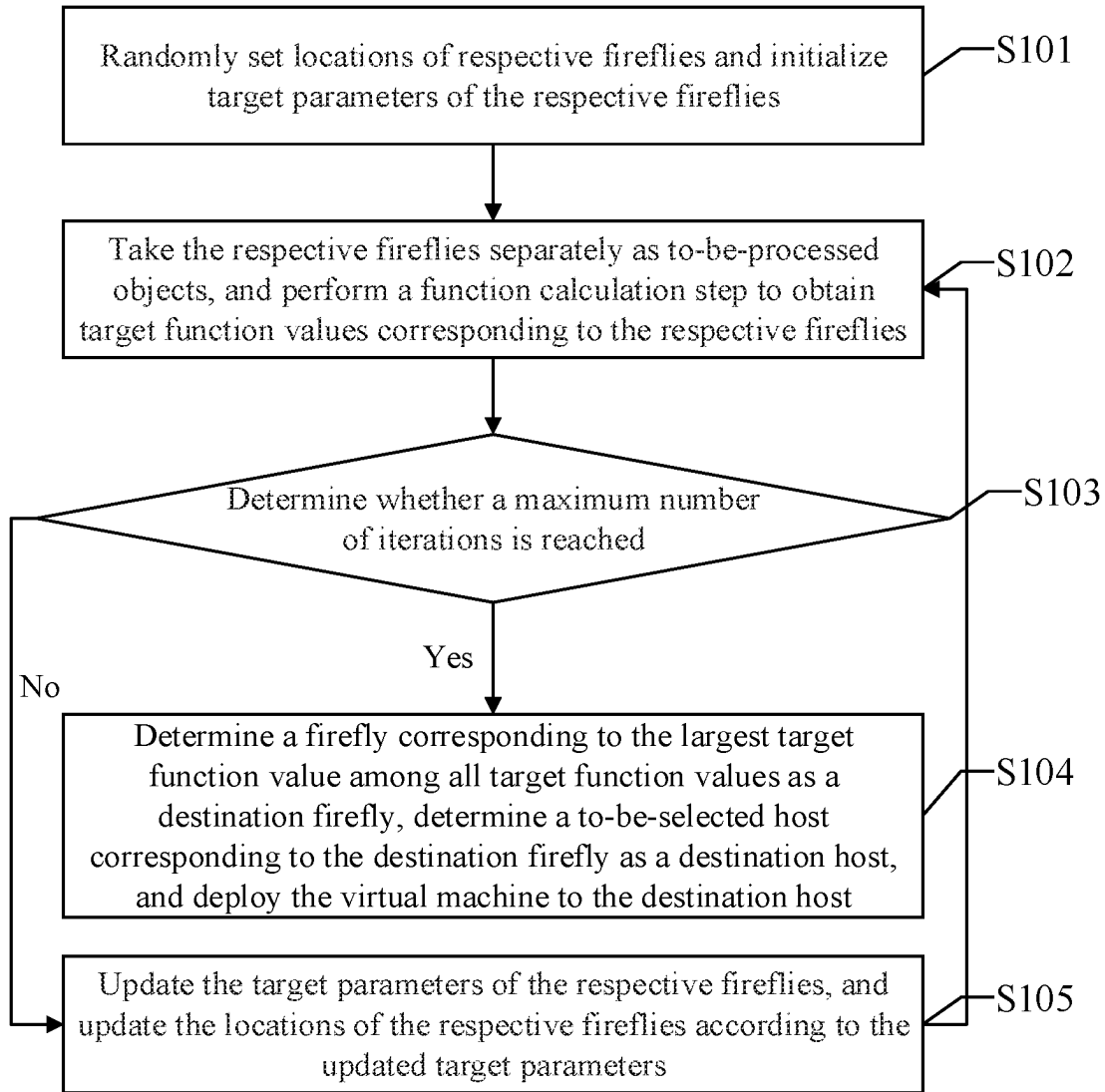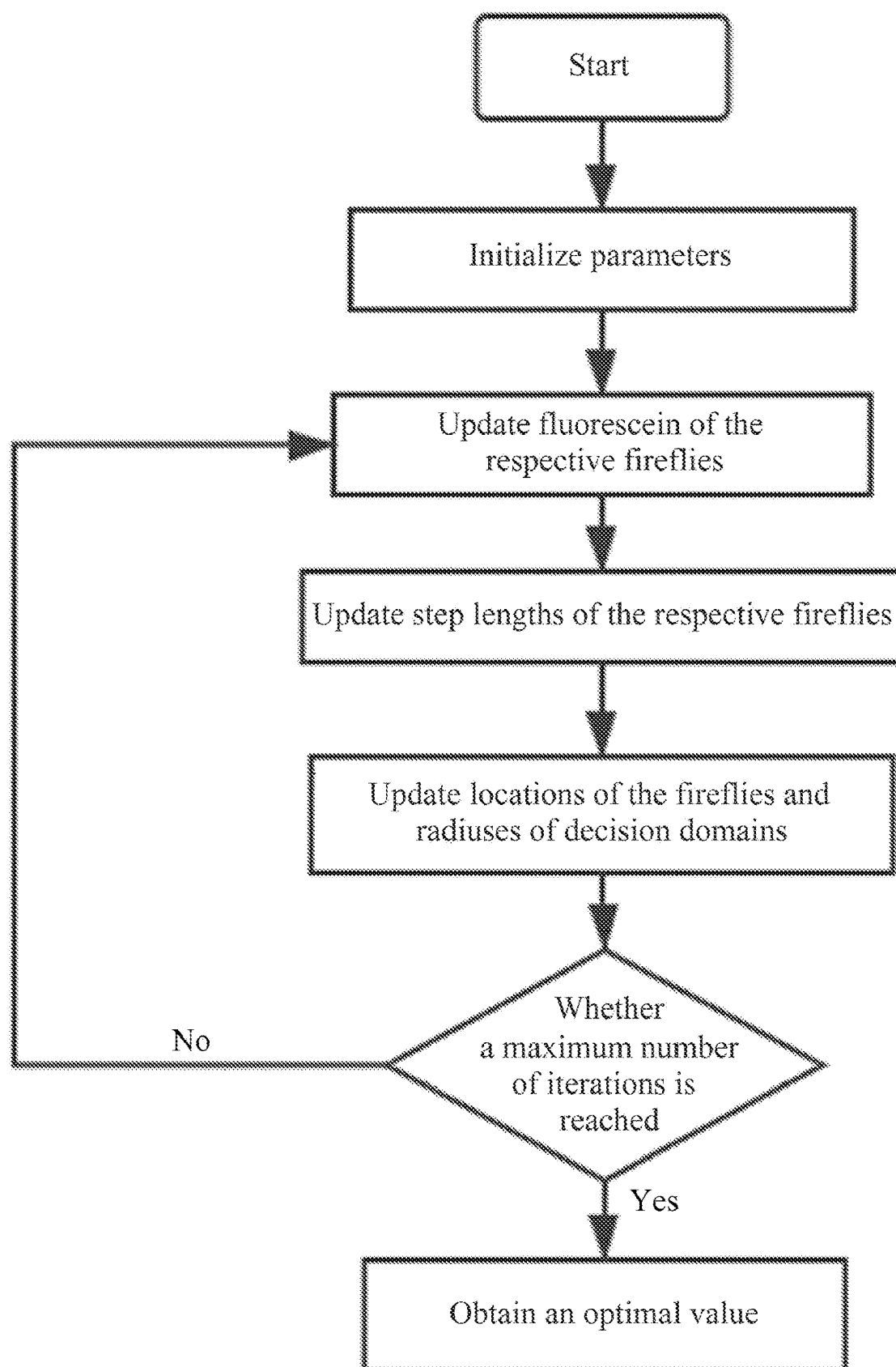
Randomly set locations of respective fireflies and initialize target parameters of the respective fireflies — S101

Take the respective fireflies separately as to-be-processed objects, and perform a function calculation step to obtain target function values corresponding to the respective fireflies — S102

Determine whether a maximum number of iterations is reached — S103

No

Yes

Determine a firefly corresponding to the largest target function value among all target function values as a destination firefly, determine a to-be-selected host corresponding to the destination firefly as a destination host, and deploy the virtual machine to the destination host — S104

Update the target parameters of the respective fireflies, and update the locations of the respective fireflies according to the updated target parameters — S105

FIG. 1

Start

Initialize parameters

Update fluorescein of the respective fireflies

Update step lengths of the respective fireflies

Update locations of the fireflies and radiuses of decision domains

Whether a maximum number of iterations is reached

No

Yes

Obtain an optimal value

FIG. 2

Initialization Module ⎯301

Function calculation module ⎯302

Determination module ⎯303

Deployment Module ⎯304
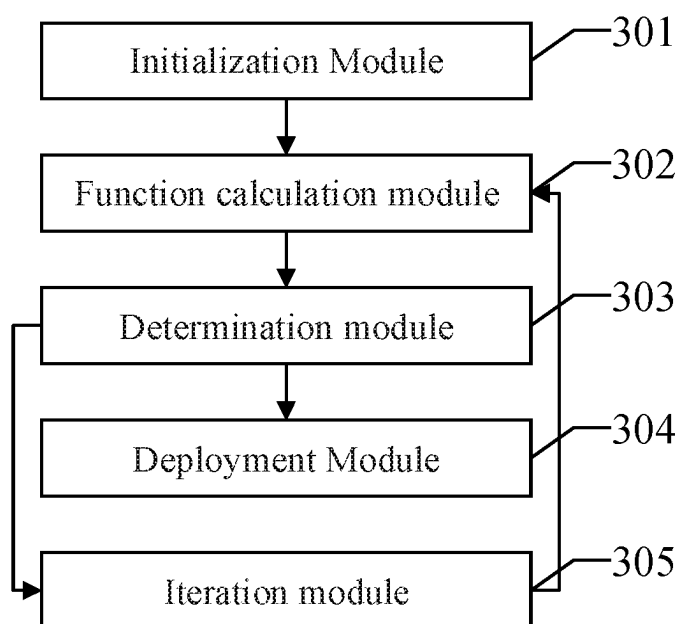
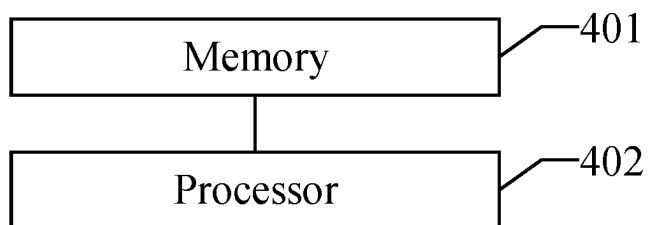Iteration module ⎯305

FIG. 3

Memory ⎯401

Processor ⎯402

FIG. 4

VIRTUAL MACHINE DEPLOYMENT METHOD AND APPARATUS, DEVICE, AND READABLE STORAGE MEDIUM

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to Chinese Patent Application No. 202011261567.6 filed on Nov. 12, 2020 and entitled "VIRTUAL MACHINE DEPLOYMENT METHOD AND APPARATUS, DEVICE, AND READABLE STORAGE MEDIUM", the entire contents of which are incorporated herein by reference.

TECHNICAL FIELD

[0002] The present application relates to the technical field of computers, and particularly relates to a virtual machine deployment method, apparatus and device, and a readable storage medium.

BACKGROUND

[0003] Virtual machines in a cloud platform can be scheduled on different hosts. Generally, a less loaded host may be selected for deployment of a new virtual machine, or a virtual machine on a highly loaded host is transferred to a less loaded host, thereby balancing the load of the individual hosts as much as possible. Although scheduling the virtual machine in such a way can avoid overload or underload of the single host, it is difficult to guarantee the operation performance of all hosts in the cloud platform as it does not take into account the public service capability of the whole cloud platform.

[0004] Thus, how to maximize the operation performance of all hosts in the cloud platform while scheduling the virtual machines is a problem that needs to be addressed by those skilled in the art.

SUMMARY

[0005] In view of this, the present application aims at providing a virtual machine deployment method, apparatus and device, and a readable storage medium to maximize the operation performance of all hosts in a cloud platform while scheduling virtual machines. The specific solution is as follows.

[0006] In a first aspect, the present application provides a virtual machine deployment method, including:

[0007] randomly setting locations of respective fireflies and initializing target parameters of the respective fireflies, wherein the locations of the respective fireflies are locations of respective to-be-selected hosts;

[0008] taking the respective fireflies separately as to-be-processed objects, and performing a function calculation step to obtain target function values corresponding to the respective fireflies; the function calculation step including: calculating a target function according to the location of each of the to-be-processed objects, wherein the target function can represent an average performance score of all the to-be-selected hosts after assuming that a virtual machine has been deployed to each of the to-be-processed objects;

[0009] determining whether a maximum number of iterations is reached;

[0010] if the maximum number of iterations is reached, determining a firefly corresponding to the largest target

function value among all target function values as a destination firefly, determining a to-be-selected host corresponding to the destination firefly as a destination host, and deploying the virtual machine to the destination host; and

[0011] if the maximum number of iterations is not reached, updating the target parameters of the respective fireflies, updating the locations of the respective fireflies according to the updated target parameters, and re-performing the step of taking the respective fireflies separately as the to-be-processed objects and performing the function calculation step to obtain target function values corresponding to the respective fireflies.

[0012] In some embodiments, the calculation formula of the target function is:

$$F =$$

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

[0013] wherein F is the target function; f_p is an average power consumption of the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{CPU} is an average CPU utilization rate of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{load} is an average resource balance degree of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; and K_1, K_2 and K_3 are preset weight values, respectively.

[0014] In some embodiments, the average resource balance degree is calculated based on the average CPU utilization rate, an average memory utilization rate, an average disk utilization rate, and an average bandwidth utilization rate of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; and the calculation formula of the average resource balance degree is:

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

[0015] wherein f_{load} is the average resource balance degree, u_{cpu} is the average CPU utilization rate, u_{mem} is the average memory utilization rate, u_{hw} is the average bandwidth utilization rate, and u_{disk} is the average disk utilization rate.

[0016] In some embodiments, the target parameter includes fluorescein and a step length. The step of updating the target parameters of the respective fireflies includes:

[0017] updating fluorescein of the respective fireflies with a first formula, wherein the first formula is:

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

[0018] updating step lengths of the respective fireflies with a second formula, wherein the second formula is:

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

[0019] wherein $l_i(t+1)$ is fluorescein of a firefly i at the $(t+1)^{th}$ iteration, $\rho(0<\rho<1)$ is a fluorescein volatilization speed of the firefly i, $l_i(t)$ is fluorescein of the firefly i at the tth iteration, γ is a fluorescein update rate of the firefly i, $F_i(t+1)$ is a target function value of the firefly i at the $(t+1)^{th}$ iteration, $s_i(t+1)$ is a step length of the firefly i at the $(t+1)^{th}$ iteration, s_{min} is a preset minimum step length, s_{max} is a preset maximum step length, t+1 is the current number of iterations, and t_{max} is the maximum number of iterations.

[0020] In some embodiments, the step of updating the locations of the respective fireflies according to the updated target parameters includes:

[0021] updating the locations of the respective fireflies with a third formula, wherein the third formula is:

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

[0022] wherein $x_i(t+1)$ is a location of the firefly i at the $(t+1)^{th}$ iteration, $x_i(t)$ is a location of the firefly i at the tth iteration, $s_i(t+1)$ is the step length of the firefly i at the $(t+1)^{th}$ iteration, $x_j(t)$ is a location of a firefly j at the tth iteration, and $\|x_j(t)-x_i(t)\|$ is a Euclidean distance between the location of the firefly i and the location of the firefly j; and

[0023] wherein the firefly j is a neighbor firefly selected within a decision domain of the firefly i at the tth iteration based on the roulette rule, and the fluorescein of the firefly j is greater than the fluorescein of the firefly i.

[0024] In some embodiments, the target parameter further includes a decision domain. Decision domains of the respective fireflies are updated with a fourth formula, wherein the fourth formula is:

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

[0025] wherein $r_d^i(t+1)$ is a decision domain of the firefly i at the $(t+1)^{th}$ iteration, r_s is a preset perceptual domain, $r_d^i(t)$ is a decision domain of the firefly i at the tth iteration, β is a dynamic decision domain update rate, n_i is a neighborhood set threshold, $N_i(t)$ is a neighborhood set of the firefly i at the tth iteration.

[0026] In some embodiments, normalization is performed on the average power consumption, the average CPU utilization rate, and the average resource balance degree, wherein a normalization formula is:

$$②$$

[0027] wherein f_p is the average power consumption after normalization, f_{p_o} is the average power consumption before normalization, f_{p_min} is a minimum average power consumption, and f_{p_max} is a maximum average power consumption; f_{cpu} is the average CPU utilization rate after normalization, f_{cpu_o} is the average CPU utilization rate before normalization, f_{cpu_min} is a minimum average CPU utilization rate, f_{cpu_max} is a maximum average CPU utilization rate; f_{load} is the average resource balance degree after normalization, f_{load_o} is the average resource balance degree before normalization, f_{load_min} is a minimum average resource balance degree, and f_{load_max} is a maximum average resource balance degree.

[0028] In a second aspect, the present application provides a virtual machine deployment apparatus, including:

[0029] an initialization module, configured to randomly set locations of respective fireflies and initialize target parameters of the respective fireflies, wherein the locations of the respective fireflies are locations of respective to-be-selected hosts;

[0030] a function calculation module, configured to take the respective fireflies separately as to-be-processed objects, and perform a function calculation step to obtain target function values corresponding to the respective fireflies; the function calculation step including: calculating a target function according to the location of each of the to-be-processed objects, wherein the target function can represent an average performance score of all the to-be-selected hosts after assuming that a virtual machine has been deployed to each of the to-be-processed objects;

[0031] a determination module, configured to determine whether a maximum number of iterations is reached;

[0032] a deployment module, configured to, if the maximum number of iterations is reached, determine a firefly corresponding to the largest target function value among all the target function values as a destination firefly, determine a to-be-selected host corresponding to the destination firefly as a destination host, and deploy the virtual machine to the destination host; and

[0033] an iteration module, configured to, if the maximum number of iterations is not reached, update the target parameters of the respective fireflies, update the locations of the respective fireflies according to the updated target parameters, and re-perform the step of taking the respective fireflies separately as the to-be-processed objects and performing the function calculation step to obtain target function values corresponding to the respective fireflies.

[0034] In a third aspect, the present application provides a virtual machine deployment device, including:

[0035] a memory, configured to store a computer program; and

[0036] a processor, configured to execute the computer program to implement the virtual machine deployment method disclosed above.

[0037] In a fourth aspect, the present application provides a readable storage medium, configured to store a computer program, wherein the computer program, when executed by a processor, implements the virtual machine deployment method disclosed above.

[0038] It can be from the above solution that the present application provides a virtual machine deployment method,

including: randomly setting locations of respective fireflies and initializing target parameters of the respective fireflies, wherein the locations of the respective fireflies are locations of respective to-be-selected hosts; taking the respective fireflies separately as to-be-processed objects, and performing a function calculation step to obtain target function values corresponding to the respective fireflies; the function calculation step including: calculating a target function according to the location of each of the to-be-processed objects, wherein the target function can represent an average performance score of all the to-be-selected hosts after assuming that a virtual machine has been deployed to each of the to-be-processed objects; determining whether a maximum number of iterations is reached; if the maximum number of iterations is reached, determining a firefly corresponding to the largest target function value among all target function values as a destination firefly, determining a to-be-selected host corresponding to the destination firefly as a destination host, and deploying the virtual machine to the destination host; and if the maximum number of iterations is not reached, updating the target parameters of the respective fireflies, updating the locations of the respective fireflies according to the updated target parameters, and re-performing the step of taking the respective fireflies separately as the to-be-processed objects and performing the function calculation step to obtain target function values corresponding to the respective fireflies.

[0039] As can be seen, according to the present application, a firefly algorithm is utilized to determine deployment locations of virtual machines, the average performance score of all the to-be-selected hosts is taken as the target function, and the locations of the respective fireflies are locations of the respective to-be-selected hosts. Therefore, an iterative optimization process of the firefly algorithm is as follows: finding a host capable of maximizing the operation performance of all the hosts in the cloud platform after the virtual machine is deployed. Since the target function is the average performance score of all the to-selected hosts after the virtual machines are deployed, the to-be-selected host corresponding to the maximum target function value is selected to maximize the average performance of all the to-be-selected hosts after the virtual machine has deployed to the destination host, thereby maximizing the operation performance of all hosts in the cloud platform while scheduling the virtual machines.

[0040] Accordingly, the present application provides a virtual machine deployment apparatus and device, and a readable storage medium, which also have the above technical effects

BRIEF DESCRIPTION OF THE DRAWINGS

[0041] In order to illustrate more clearly the technical solutions in the embodiments of the present application or in the prior art, the drawings required for the description of the embodiments or prior art will now be briefly described below. Obviously, the drawings in the following description are merely embodiments of the present application, and a person of ordinary skill in the art may also obtain other drawings according to the provided drawings without inventive step.

[0042] FIG. **1** is a flowchart illustrating a virtual machine deployment method according to the present application;

[0043] FIG. **2** is a flowchart illustrating a method for solving a target function using a firefly algorithm according to the present application;

[0044] FIG. **3** is a schematic diagram of a virtual machine deployment apparatus according to the present application; and

[0045] FIG. **4** is a schematic diagram of a virtual machine deployment device according to the present application.

DETAILED DESCRIPTION

[0046] The technical solutions in the embodiments of the present application will be clearly and completely described below in conjunction with drawings in the embodiments of the present application. Obviously, the described embodiments are only a part of the embodiments of the present application, not all of the embodiments. Based on the embodiments in the present application, all other embodiments acquired by a person of ordinary skill in the art without creative work shall fall within the protection scope of the present application.

[0047] Prior to the introduction of the present application, the relevant background art is introduced as follows.

[0048] In general, virtual machines need to be scheduled in the following three scenarios. I: A user creates a new virtual machine, and selects a physical host to which the virtual machine is deployed. II: If there are excessive virtual machines on one physical host to result in overload, part of the virtual machines on the physical host need to be transferred to other physical hosts. Alternatively, if there are too few virtual machines on one physical host to result in underload, virtual machines on other physical hosts need to be transferred to the current physical host in order to achieve load balance. III: If the physical host fails, virtual machines on the physical host need to be transferred to other physical hosts. Unreasonable scheduling strategies may cause multiple transfers of virtual machines across physical hosts, severely impacting the stability of the entire cloud platform. Moreover, the resources and applications requested by each user using the cloud platform are different, and loads on the respective physical hosts are also different, resulting in instability in resource usage by the cloud platform.

[0049] At present, an ant colony optimization algorithm can be used to find a target host for deploying the virtual machine. However, the anti colony optimization algorithm has the problems of low solution speed and the difficulty in ensuring the quality of the obtained solution, the calculation amount is large, the convergence speed is low; and it is easy to fall into local optimization, which cannot maximize the operation performance of all hosts in the cloud platform while scheduling the virtual machines. To this end, the present application provides a virtual machine deployment solution that maximizes the operation performance of all hosts in a cloud platform while scheduling virtual machines.

[0050] Referring to FIG. **1**, an embodiment of the present application discloses a virtual machine deployment method including the following steps.

[0051] S101, locations of respective fireflies are randomly set, and target parameters of the respective fireflies are initialized.

[0052] The locations of the respective fireflies are the locations of respective to-be-selected hosts. The quantity of the locations of the fireflies is equal to the quantity of to-be-selected hosts, but the quantity of the fireflies and the quantity of the to-be-selected hosts may be equal or may not

be equal. If the quantity of the fireflies is greater than the quantity of the to-be-selected hosts, a plurality of fireflies may be gathered at one location of the firefly.

[0053] S102, the respective fireflies are separately taken as to-be-processed objects, and a function calculation step is performed to obtain target function values corresponding to the respective fireflies.

[0054] The function calculation step includes: calculating a target function according to the location of each of the to-be-processed objects, wherein the target function can represent an average performance score of all the to-be-selected hosts after assuming that a virtual machine has been deployed to each of the to-be-processed objects.

[0055] In a specific embodiment, the calculation formula of the target function is:

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

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

[0056] wherein F is the target function; f_p is an average power consumption of the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{CPU} is an average CPU utilization rate of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{load} is an average resource balance degree of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; and K_1, K_2 and K_3 are preset weight values, respectively. The greater preset weight value indicates that the cloud platform currently focuses on its corresponding parameter (power consumption, CPU utilization rate, or resource balance degree).

[0057] For example, it is considered that a firefly A corresponds to a to-be-selected host A, and assuming that a virtual machine is deployed to the to-be-selected host A, the power consumption, CPU utilization rate and resource balance degree of the to-be-selected host A are adjusted based on the power consumption, CPU utilization rate, bandwidth utilization rate, memory utilization rate, and other relevant parameters of the virtual machine. Then, the average power consumption, average CPU utilization rate, and average resource balance degree of all the to-be-selected hosts (including the to-be-selected host A to which the virtual machine is deployed) are calculated. The greater F indicates better average performance of all the to-be-selected hosts.

[0058] In a specific embodiment, the average resource balance degree is calculated based on the average CPU utilization rate, an average memory utilization rate, an average disk utilization rate, and an average bandwidth utilization rate of all the to-be-selected hosts at a certain moment after assuming that the virtual machine has been deployed to each of the to-be-processed objects. The calculation formula of the average resource balance degree is:

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

[0059] wherein f_{load} is the average resource balance degree, u_{cpu} is the average CPU utilization rate, u_{mem} is the average memory utilization rate, u_{hw} is the average bandwidth utilization rate, and u_{disk} is the average disk utilization rate.

[0060] In a specific embodiment, normalization is performed on the average power consumption, the average CPU utilization rate, and the average resource balance degree, wherein a normalization formula is:

⑦

[0061] wherein f_p is the average power consumption after normalization, f_{p_o} is the average power consumption before normalization, f_{p_min} is a minimum average power consumption, and f_{p_max} is a maximum average power consumption; f_{cpu} is the average CPU utilization rate after normalization, f_{cpu_o} is the average CPU utilization rate before normalization, f_{cpu_min} is a minimum average CPU utilization rate, f_{cpu_max} is a maximum average CPU utilization rate; f_{load} is the average resource balance degree after normalization, f_{load_o} is the average resource balance degree before normalization, f_{load_min} is a minimum average resource balance degree, and f_{load_max} is a maximum average resource balance degree. The average power consumption, average CPU utilization rate, and average resource balance degree after normalization take values between 0 and 1.

[0062] S103, whether a maximum number of iterations is reached is determined; if the maximum number of iterations is reached, S104 is performed; and if the maximum number of iterations is not reached, S105 is performed.

[0063] S104, a firefly corresponding to the largest target function value among all target function values is determined as a destination firefly, a to-be-selected host corresponding to the destination firefly is determined as a destination host, and the virtual machine is deployed to the destination host.

[0064] S105, the target parameters of the respective fireflies are updated, the locations of the respective fireflies are updated according to the updated target parameters, and S102 is re-performed.

[0065] In a specific embodiment, the target parameter includes fluorescein and a step length. The step that the target parameters of the respective fireflies are updated includes:

[0066] updating fluorescein of the respective fireflies with a first formula, wherein the first formula is:

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

[0067] updating step lengths of the respective fireflies with a second formula, wherein the second formula is:

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

[0068] wherein $l_i(t+1)$ is fluorescein of a firefly i at the $(t+1)^{th}$ iteration, i.e., fluorescein updated currently; $\rho(0<\rho<1)$ is a fluorescein volatilization speed of the firefly i; $l_i(t)$ is fluorescein of the firefly i at the tth iteration; γ is a fluorescein update rate of the firefly i; $F_i(t+1)$ is a target function value of the firefly i at the $(t+1)^{th}$ iteration; $s_i(t+1)$ is a step length of the firefly i at the $(t+1)^{th}$ iteration, i.e, a step length updated currently; s_{min} is a preset minimum step length that takes a value close to 0; s_{max} is a preset maximum step length that takes a value close to 1; t+1 is the current number of iterations; and t_{max} is the maximum number of iterations.

[0069] It is noted that the step length of each of the fireflies varies with increasing number of iterations, thereby ensuring that the step length of each of the fireflies remains within a reasonable range.

[0070] In a specific embodiment, the step that the locations of the respective fireflies are updated according to the updated target parameters includes:

[0071] updating the locations of the respective fireflies with a third formula, wherein the third formula is:

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

[0072] wherein $x_i(t+1)$ is a location of the firefly i at the $(t+1)^{th}$ iteration, $x_i(t)$ is a location of the firefly i at the tth iteration, $s_i(t+1)$ is the step length of the firefly i at the $(t+1)^{th}$ iteration, $x_j(1)$ is a location of a firefly j at the tth iteration, and $\|x_j(t)-x_i(t)\|$ is a Euclidean distance between the location of the firefly i and the location of the firefly j; and

[0073] wherein the firefly j is a neighbor firefly selected within a decision domain of the firefly i at the tth iteration based on the roulette rule, and the fluorescein of the firefly j is greater than the fluorescein of the firefly i.

[0074] In a specific embodiment, the target parameter further includes a decision domain. Decision domains of the respective fireflies are updated with a fourth formula, wherein the fourth formula is:

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

[0075] wherein $r_d{}^i(t+1)$ is a decision domain of the firefly i at the $(t+1)^{th}$ iteration, r_s is a preset perceptual domain, $r_d{}^i(t)$ is a decision domain of the firefly i at the tth iteration, β is a dynamic decision domain update rate, n_i is a neighborhood set threshold, and $N_i(t)$ is a neighborhood set of the firefly i at the tth iteration, i.e, the quantity of neighbor fireflies in the decision domain of the firefly i at the tth iteration.

[0076] As can be seen, according to the embodiments of the present application, a firefly algorithm is utilized to determine deployment locations of the virtual machines, the average performance score of all the to-be-selected hosts is taken as the target function, and the locations of the respective fireflies are locations of the respective to-be-selected hosts. Therefore, an iterative optimization process of the firefly algorithm is as follows: finding a host capable of maximizing the operation performance of all the hosts in the cloud platform after the virtual machine is deployed. Since the target function is the average performance score of all the to-selected hosts after the virtual machines are deployed, the to-be-selected host corresponding to the maximum target function value is selected to maximize the average performance of all the to-be-selected hosts after the virtual machine has deployed to the destination host, thereby maximizing the operation performance of all hosts in the cloud platform while scheduling the virtual machines.

[0077] An embodiment of the present application discloses a virtual machine deployment solution, including defining a target function, solving a target function, and deploying virtual machines.

[0078] 1. Assuming that a virtual machine has been deployed to each of to-be-selected hosts, a target function F is defined:

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

[0079] wherein f_p is an average power consumption of the respective to-be-selected hosts; f_{CPU} is an average CPU utilization rate of all the to-be-selected hosts; f_{load} is an average resource balance degree of all the to-be-selected hosts; and K_1, K_2 and K_3 $_a$re preset weight values, respectively.

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

[0080] wherein f_{load} is the average resource balance degree, u_{cpu} is the average CPU utilization rate, u_{mem} is an average memory utilization rate, u_{hw} is an average bandwidth utilization rate, and u_{disk} is an average disk utilization rate.

[0081] f_p, f_{cpu} and f_{load} are all values after normalization and a normalization formula is:

$$⑦$$

⑦ indicates text missing or illegible when filed

[0082] wherein f_p is the average power consumption after normalization, f_{p_o} is the average power consumption before normalization, f_{p_min} is a minimum average

power consumption, and f_{p_max} is a maximum average power consumption; f_{cpu} is the average CPU utilization rate after normalization, f_{cpu_o} is the average CPU utilization rate before normalization, f_{cpu_min} is a minimum average CPU utilization rate, f_{cpu_max} is a maximum average CPU utilization rate; f_{load} is the average resource balance degree after normalization, f_{load_o} is the average resource balance degree before normalization, f_{load_min} is a minimum average resource balance degree, and f_{load_max} is a maximum average resource balance degree.

[0083] 2. Referring to FIG. 2, the target function is solved using the firefly algorithm.

[0084] (1) Parameters are initialized.

[0085] Locations of respective fireflies are initialized; and the locations of the respective fireflies are the locations of the respective to-be-selected hosts.

[0086] Target parameters, such as fluorescein, step length and decision domain, of the respective fireflies are initialized.

[0087] respective preset values, such as fluorescein volatilization speed $\rho(0<\rho<1)$, fluorescein update rate γ, minimum step length s_{min}, maximum step length s_{max}, maximum number of iterations t_{max}, perceptual domain r_s, neighborhood set threshold n_t, are set.

[0088] (2) Assuming that a virtual machine has been deployed to each of the to-be-selected hosts respectively, target function values corresponding to the respective fireflies are calculated based on the current locations of the fireflies. The larger the target function values, the better the average performance of all the to-be-selected hosts.

[0089] (3) Fluorescein of the respective fireflies is updated.

[0090] The fluorescein of the firefly is directly related to its location. The greater the fluorescein, the higher the fitness value of its location. The size of the fluorescein is also related to the size of the fluorescein in the previous iteration and its volatilization speed, as well as the target function value, wherein an update formula is:

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

[0091] wherein $l_i(t+1)$ is fluorescein of a firefly i at the $(t+1)^{th}$ iteration, i.e., fluorescein updated currently; $\rho(0<\rho<1)$ is a fluorescein volatilization speed of the firefly i; $l_i(t)$ is fluorescein of the firefly i at the tth iteration; γ is a fluorescein update rate of the firefly i; and $F_i(t+1)$ is a target function value of the firefly i at the $(t+1)^{th}$ iteration.

[0092] (4) Step lengths of the respective fireflies are updated.

[0093] The step length remains greater at the initial stage of iteration in the algorithm. In order to avoid falling into local optimization due to premature maturation of the algorithm, at the later stage of iteration in the algorithm, the step length is gradually reduced to avoid skipping the optimal solution at the later stage of iteration in the algorithm due to the excessively greater step length. A step length update formula is:

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

[0094] wherein $s_i(t+1)$ is a step length of the firefly i at the $(t+1)^{th}$ iteration, i.e, a step length updated currently; s_{min} is a preset minimum step length that takes a value close to 0; s_{max} is a preset maximum step length that takes a value close to 1; $l_i(t)$ is fluorescein of the firefly i at the tth iteration; t+1 is the current number of iterations; and t_{max} is the maximum number of iterations.

[0095] (5) The locations of the respective fireflies are updated.

[0096] An eligible neighbor firefly needs to be found first before each of the fireflies moves according to the step length. The condition for finding the neighbor firefly is as follows: the location of the neighbor firefly is within the decision domain of the current firefly, and the fluorescein of the neighbor firefly is greater than the fluorescein of the current firefly. Neighbor fireflies satisfying the above condition are combined to form a neighborhood set, and the probability of each neighbor firefly in the neighborhood set is calculated according to the following formula:

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

[0097] wherein $j \in N_i(t)$, $N_i(t) = \{j; d_{i,j}(t) < r_d^i(t); l_i(t) < l_j(t)\}$, $N_i(t)$ is the neighborhood set of any firefly i at the tth iteration, $d_{i,j}(t) = \|x_j(t-x_i(t)\|$ is a Euclidean distance between the firefly i and a firefly j at the tth iteration.

[0098] Based on the probability of each neighbor firefly, a neighbor firefly j is selected using the roulette rule, and then the spatial location of the firefly i is updated according to the following formula:

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

[0099] wherein $x_i(t+1)$ is a location of the firefly i at the $(t+1)^{th}$ iteration, $x_i(t)$ is a location of the firefly i at the tth iteration, $s_i(t+1)$ is the step length of the firefly i at the $(t+1)^{th}$ iteration, $x_i(t)$ is a location of a firefly j at the tth iteration, and $\|x_j(t) - x_i(t)\|$ is a Euclidean distance between the location of the firefly i and the location of the firefly j.

[0100] (6) Decision domains of the respective fireflies are updated.

[0101] After the locations of the fireflies are updated, the decision domains of the fireflies are further updated, wherein an update formula is:

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

[0102] wherein $r_d^i(t+1)$ is a decision domain of the firefly i at the $(t+1)^{th}$ iteration, r_s is a preset perceptual

domain, $r_d{}^i(t)$ is a decision domain of the firefly i at the tth iteration, β is a dynamic decision domain update rate, n_i is a neighborhood set threshold, and $N_i(t)$ is a neighborhood set of the firefly i at the tth iteration, i.e. the quantity of neighbor fireflies in the decision domain of the firefly i at the tth iteration.

[0103] (7) The number of iterations at the moment is determined.

[0104] If the current number of iterations is greater than the maximum number of iteration/max. iteration ends, the target function values are calculated using the latest locations of the fireflies, and the maximum target function value and the corresponding firefly location are selected for output. If the current number of iterations is less than the maximum number of iterations t_{max}, step (2) is re-performed for the next iteration until the constraint on the number of iterations is satisfied.

[0105] 3. A virtual machine is deployed: the location of the firefly corresponding to the maximum target function value corresponds is the deployment location of the virtual machine, wherein the virtual machine is deployed to the location, which maximizes the operation performance of all the to-be-selected hosts.

[0106] As can be seen, in the embodiment, the target function capable of indicating the average performance score of all the to-be-selected hosts is defined by taking into account the resource balance degree obtained based on the CPU utilization rate, the memory utilization rate, the bandwidth utilization rate, and the disk utilization rate, in conjunction with normalization of the CPU utilization rate and the power consumption. Solving the target function using a modified firefly algorithm takes into account the influence of the size of fluorescein as well as the step length on the iteration process, such that the algorithm has strong local search capability, can find the optimal solution for that region within a smaller region, is convenient to operate, and easy to implement. Moreover, since there are parameters for the firefly algorithm, the influence of the parameters on the algorithm is slight. The firefly algorithm used by the embodiment can present the individual from oscillation near the peak, and an optimal physical host location is output after iterative optimization to serve as the destination physical host of the virtual machine, such that the cloud platform can maintain high CPU utilization rate, lower energy consumption, and resource balance after the virtual machine is deployed to improve overall system performance.

[0107] A virtual machine deployment apparatus provided by an embodiment of the present application will be described below; which can refer to the virtual machine deployment method described above.

[0108] Referring to FIG. 3, an embodiment of the present application discloses a virtual machine deployment apparatus, including:

[0109] an initialization module 301, configured to randomly set locations of respective fireflies and initialize target parameters of the respective fireflies, wherein the locations of the respective fireflies are locations of respective to-be-selected hosts;

[0110] a function calculation module 302, configured to take the respective fireflies separately as to-be-processed objects, and perform a function calculation step to obtain target function values corresponding to the respective fireflies; the function calculation step including: calculating a target function according to the

location of each of the to-be-processed objects, wherein the target function can represent an average performance score of all the to-be-selected hosts after assuming that a virtual machine has been deployed to each of the to-be-processed objects;

[0111] a determination module 303, configured to determine whether a maximum number of iterations is reached;

[0112] a deployment module 304, configured to, if the maximum number of iterations is reached, determine a firefly corresponding to the largest target function value among all the target function values as a destination firefly, determine a to-be-selected host corresponding to the destination firefly as a destination host, and deploy the virtual machine to the destination host; and

[0113] an iteration module 305, configured to, if the maximum number of iterations is not reached, update the target parameters of the respective fireflies, update the locations of the respective fireflies according to the updated target parameters, and re-perform the step of taking the respective fireflies separately as the to-be-processed objects and performing the function calculation step to obtain target function values corresponding to the respective fireflies.

[0114] In a specific embodiment, the calculation formula of the target function is:

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

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

[0115] wherein F is the target function; f_p is an average power consumption of the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{CPU} is an average CPU utilization rate of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{load} is an average resource balance degree of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; and K_1, K_2 and K_3 are preset weight values, respectively.

[0116] In a specific embodiment, the average resource balance degree is calculated based on the average CPU utilization rate, an average memory utilization rate, an average disk utilization rate, and an average bandwidth utilization rate of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects. The calculation formula of the average resource balance degree is:

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

[0117] wherein f_{load} is the average resource balance degree, u_{cpu} is the average CPU utilization rate, u_{mem} is

the average memory utilization rate, u_{hw} is the average bandwidth utilization rate, and u_{disk} is the average disk utilization rate.

[0118] In a specific embodiment, the target parameter includes fluorescein and a step length, and the iteration module is specifically configured to:

[0119] update fluorescein of the respective fireflies with a first formula, wherein the first formula is:

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

[0120] update step lengths of the respective fireflies with a second formula, wherein the second formula is:

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

[0121] wherein $l_i(t+1)$ is fluorescein of a firefly i at the $(t+1)^{th}$ iteration, $\rho(0<\rho<1)$ is a fluorescein volatilization speed of the firefly i, $l_i(t)$ is fluorescein of the firefly i at the tth iteration, γ is a fluorescein update rate of the firefly i, $F_i(t+1)$ is a target function value of the firefly i at the $(t+1)^{th}$ iteration, $s_i(t+1)$ is a step length of the firefly i at the $(t+1)^{th}$ iteration, s_{min} is a preset minimum step length, s_{max} is a preset maximum step length, t+1 is the current number of iterations, and t_{max} is the maximum number of iterations.

[0122] In a specific embodiment, the iteration module is specifically configured to:

[0123] update the locations of the respective fireflies with a third formula, wherein the third formula is:

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

[0124] wherein $x_i(t+1)$ is a location of the firefly i at the $(t+1)^{th}$ iteration, $x_i(t)$ is a location of the firefly i at the tth iteration, $s_i(t+1)$ is the step length of the firefly i at the $(t+1)^{th}$ iteration, $x_j(t)$ is a location of a firefly j at the tth iteration, and $\|x_i(t)-x_i(t)\|$ is a Euclidean distance between the location of the firefly i and the location of the firefly j; and

[0125] wherein the firefly j is a neighbor firefly selected within a decision domain of the firefly i at the tth iteration based on the roulette rule, and the fluorescein of the firefly j is greater than the fluorescein of the firefly i.

[0126] In a specific embodiment, the target parameter further includes a decision domain. Decision domains of the respective fireflies are updated with a fourth formula, wherein the fourth formula is:

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

[0127] wherein $r_d^i(t+1)$ is a decision domain of the firefly i at the $(t+1)^{th}$ iteration, r_s is a preset perceptual domain, $r_d^i(t)$ is a decision domain of the firefly i at the

t^{th} iteration, β is a dynamic decision domain update rate, n_i is a neighborhood set threshold, $N_i(t)$ is a neighborhood set of the firefly i at the tth iteration.

[0128] In a specific embodiment, normalization is performed on the average power consumption, the average CPU utilization rate, and the average resource balance degree, wherein a normalization formula is:

$$\begin{cases} f_p = 1 - \dfrac{f_{p\_o} - f_{p\_min}}{f_{p\_max} - f_{p\_min}} \\ f_{cpu} = \dfrac{f_{cpu\_o} - f_{cpu\_min}}{f_{cpu\_max} - f_{cpu\_min}} \\ f_{load} = \dfrac{f_{load\_o} - f_{load\_min}}{f_{load\_max} - f_{load\_min}} \end{cases}$$

[0129] wherein f_p is the average power consumption after normalization, f_{p_o} is the average power consumption before normalization, f_{p_min} is a minimum average power consumption, and f_{p_max} is a maximum average power consumption; f_{cpu} is the average CPU utilization rate after normalization, f_{cpu_o} is the average CPU utilization rate before normalization, f_{cpu_min} is a minimum average CPU utilization rate, f_{cpu_max} is a maximum average CPU utilization rate; f_{load} is the average resource balance degree after normalization, f_{load_o} is the average resource balance degree before normalization, f_{load_min} is a minimum average resource balance degree, f_{load_max} is a maximum average resource balance degree.

[0130] The more specific operation processes of the various modules and units in the embodiment can refer to corresponding content disclosed in the preceding embodiments, which are not repeated here.

[0131] As can be seen, the virtual machine deployment apparatus provided by the embodiment can maximize the operation performance of all hosts in a cloud platform while scheduling virtual machines.

[0132] A virtual machine deployment device provided by an embodiment of the present application will be described below, which can refer to the virtual machine deployment method and apparatus described above.

[0133] Referring to FIG. 4, an embodiment of the present application discloses a virtual machine deployment device, including:

[0134] a memory 401, configured to store a computer program; and

[0135] a processor 402, configured to execute the computer program to implement the method of any of the above embodiments.

[0136] A readable storage medium provided by an embodiment of the present application will be described below, which can refer to the virtual machine deployment method, apparatus and device described above.

[0137] A readable storage medium is configured to store a computer program, wherein the computer program, when executed by a processor, implements the virtual machine deployment method disclosed by the preceding embodiments. Specific steps of the method can refer to corresponding content disclosed in the preceding embodiments, which are not repeated here.

[0138] "First," "second," "third," "fourth," or the like (if any) involved in the present application are used for distinguishing between similar objects and not used for describing

a particular sequence or precedence order. It is to be understood that data so used may be interchanged under appropriate circumstances such that the embodiments described herein can be implemented in sequence other than those illustrated or described herein. Furthermore, the terms "include", "comprise" and any variations thereof intend to cover non-exclusive inclusion, for example, a process, method or device that includes a list of steps or units is not necessarily limited to those steps or elements expressly listed but may include other steps or elements not expressly listed or inherent to such process, method or device.

[0139] It is to be noted that descriptions referring to "first", "second", etc., in the present application are for descriptive purposes only and are not to be understood as indicating or implying relative importance thereof or as implying the quantity of technical features indicated. Thus, a feature defined with "first" and "second" may explicitly or implicitly include at least one such feature. In addition, the technical solutions of the various embodiments may be combined with each other, but must be based on being implemented by a person of ordinary skill in the art. The combination of the technical solutions should be considered to be absent and not within the scope of protection claimed in the present application when such combination is contradictory or impossible to be implemented.

[0140] Various embodiments in the description are described in an incremental manner, wherein each embodiment focuses differences from the other embodiments, and the same or similar parts of the various embodiments may refer to each other.

[0141] The steps of the method or algorithm described in combination with the embodiments disclosed herein may be implemented directly by hardware, a software module executed by a processor, or a combination of the hardware and the software module. A software module may be arranged in a random access memory (RAM), a memory, a read only memory (ROM), an electrically programmable ROM, an electrically erasable programmable ROM, a register, a hard disk, a removable disk, a CD-ROM, or a readable storage medium in any other from known in the art.

[0142] The principles and embodiments of the present application are illustrated by applying examples, and the description of the above embodiments is only used to assist in understanding the method of the present application and its core concept. Meanwhile, for a person of ordinary in the art, changes may be made to the specific embodiments and application range based on the concept of the present application. In conclusion, the description should not be construed as limiting the present application.

What is claimed is:

1. A virtual machine deployment method, comprising:

randomly setting locations of respective fireflies and initializing target parameters of the respective fireflies, wherein the locations of the respective fireflies are locations of respective to-be-selected hosts;

taking the respective fireflies separately as to-be-processed objects, and performing a function calculation step to obtain target function values corresponding to the respective fireflies; the function calculation step comprising: calculating a target function according to the location of each of the to-be-processed objects, wherein the target function can represent an average performance score of all the to-be-selected hosts after

assuming that a virtual machine has been deployed to each of the to-be-processed objects;

determining whether a maximum number of iterations is reached;

when the maximum number of iterations is reached, determining a firefly corresponding to the largest target function value among all target function values as a destination firefly, determining a to-be-selected host corresponding to the destination firefly as a destination host, and deploying the virtual machine to the destination host; and

when the maximum number of iterations is not reached, updating the target parameters of the respective fireflies, updating the locations of the respective fireflies according to the updated target parameters, and re-performing the step of taking the respective fireflies separately as the to-be-processed objects and performing the function calculation step to obtain target function values corresponding to the respective fireflies.

2. The virtual machine deployment method according to claim 1, wherein the calculation formula of the target function is:

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

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

wherein F is the target function; f_p is an average power consumption of the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{CPU} is an average CPU utilization rate of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; f_{load} is an average resource balance degree of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; K_1, K_2 and K_3 are preset weight values, respectively.

3. The virtual machine deployment method according to claim 2, wherein the average resource balance degree is calculated based on the average CPU utilization rate, an average memory utilization rate, an average disk utilization rate, and an average bandwidth utilization rate of all the to-be-selected hosts after assuming that the virtual machine has been deployed to each of the to-be-processed objects; and the calculation formula of the average resource balance degree is:

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

wherein f_{load} is the average resource balance degree, u_{cpu} is the average CPU utilization rate, u_{mem} is the average memory utilization rate, u_{hw} is the average bandwidth utilization rate, and u_{disk} is the average disk utilization rate.

4. The virtual machine deployment method according to claim **1**, wherein the target parameter comprises fluorescein and a step length; the step of updating the target parameters of the respective fireflies comprises:

updating fluorescein of the respective fireflies with a first formula, wherein the first formula is:

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

updating step lengths of the respective fireflies with a second formula, wherein the second formula is:

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

wherein $l_i(t+1)$ is fluorescein of a firefly i at the $(t+1)^{th}$ iteration, $\rho(0 < \rho < 1)$ is a fluorescein volatilization speed of the firefly i, $l_i(t)$ is fluorescein of the firefly i at the tth iteration, γ is a fluorescein update rate of the firefly i, $F_i(t+1)$ is a target function value of the firefly i at the $(t+1)^{th}$ iteration, $s_i(t+1)$ is a step length of the firefly i at the $(t+1)^{th}$ iteration, s_{min} is a preset minimum step length, s_{max} is a preset maximum step length, t+1 is the current number of iterations, and t_{max} is the maximum number of iterations.

5. The virtual machine deployment method according to claim **4**, wherein the step of updating the locations of the respective fireflies according to the updated target parameters comprises:

updating the locations of the respective fireflies with a third formula, wherein the third formula is:

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

wherein $x_i(t+1)$ is a location of the firefly i at the $(t+1)^{th}$ iteration, $x_i(t)$ is a location of the firefly i at the th iteration, $s_i(t+1)$ is the step length of the firefly i at the $(t+1)^{th}$ iteration, $x_j(t)$ is a location of a firefly j at the tth iteration, and $\|x_j(t) - x_i(t)\|$ is a Euclidean distance between the location of the firefly i and the location of the firefly j; and

wherein the firefly j is a neighbor firefly selected within a decision domain of the firefly i at the tth iteration based on the roulette rule, and the fluorescein of the firefly j is greater than the fluorescein of the firefly i.

6. The virtual machine deployment method according to claim **5**, wherein the target parameter further comprises a decision domain; and decision domains of the respective fireflies are updated with a fourth formula, wherein the fourth formula is:

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

wherein $r_d^i(t+1)$ is a decision domain of the firefly i at the $(t+1)^{th}$ iteration, r_s is a preset perceptual domain, $r_d^i(t)$ is a decision domain of the firefly i at the tth iteration,

β is a dynamic decision domain update rate, n_i is a neighborhood set threshold, $N_i(t)$ is a neighborhood set of the firefly i at the tth iteration.

7. The virtual machine deployment method according to claim **3**, comprising: performing normalization on the average power consumption, the average CPU utilization rate, and the average resource balance degree, wherein a normalization formula is:

$$\begin{cases} f_p = 1 - \dfrac{f_{p\_o} - f_{p\_min}}{f_{p\_max} - f_{p\_min}} \\ f_{cpu} = \dfrac{f_{cpu\_o} - f_{cpu\_min}}{f_{cpu\_max} - f_{cpu\_min}} \\ f_{load} = \dfrac{f_{load\_o} - f_{load\_min}}{f_{load\_max} - f_{load\_min}} \end{cases}$$

wherein f_p is the average power consumption after normalization, f_{p_o} is the average power consumption before normalization, f_{p_min} is a minimum average power consumption, and f_{p_max} is a maximum average power consumption; f_{cpu} is the average CPU utilization rate after normalization, f_{cpu_o} is the average CPU utilization rate before normalization, f_{cpu_min} is a minimum average CPU utilization rate, f_{cpu_max} is a maximum average CPU utilization rate; f_{load} is the average resource balance degree after normalization, f_{load_o} is the average resource balance degree before normalization, f_{load_min} is a minimum average resource balance degree, f_{load_max} is a maximum average resource balance degree.

8. A virtual machine deployment apparatus, comprising:

an initialization module, configured to randomly set locations of respective fireflies and initialize target parameters of the respective fireflies, wherein the locations of the respective fireflies are locations of respective to-be-selected hosts;

a function calculation module, configured to take the respective fireflies separately as to-be-processed objects, and perform a function calculation step to obtain target function values corresponding to the respective fireflies; the function calculation step comprising: calculating a target function according to the location of each of the to-be-processed objects, wherein the target function can represent an average performance score of all the to-be-selected hosts after assuming that a virtual machine has been deployed to each of the to-be-processed objects;

a determination module, configured to determine whether a maximum number of iterations is reached;

a deployment module, configured to, when the maximum number of iterations is reached, determine a firefly corresponding to the largest target function value among all the target function values as a destination firefly, determine a to-be-selected host corresponding to the destination firefly as a destination host, and deploy the virtual machine to the destination host; and

an iteration module, configured to, when the maximum number of iterations is not reached, update the target parameters of the respective fireflies, update the locations of the respective fireflies according to the updated target parameters, and re-perform the step of taking the respective fireflies separately as the to-be-processed

objects and performing the function calculation step to obtain target function values corresponding to the respective fireflies.

9. A virtual machine deployment device, comprising:

a memory, configured to store a computer program; and

a processor, configured to execute the computer program to implement the virtual machine deployment method according to any one of claims **1-7**.

10. A readable storage medium, configured to store a computer program, wherein the computer program, when executed by a processor, implements the virtual machine deployment method according to any one of claims **1-7**.

* * * * *