Virtual Machine Optimization Scheduling Analysis Based on Cloud Computing in High-Performance Computing Platforms

Xionggang Yang

Guangdong Mechanical & Electrical Polytechnic, College of Computer Applications, Guangzhou 510440, China.

Abstract: The rational allocation of virtual machine resources in cloud computing is a critical technical issue, and the efficient utilization of virtual machine resources is a key factor influencing cloud service performance. Based on this, this paper proposes a novel ant colony optimization method that can effectively address the resource scheduling problem in cloud platforms.

Keywords: Cloud Computing; Virtual Machine; Ant Colony

1. Cloud Computing Overview

Cloud computing is a distributed computing paradigm based on virtualization technology, utilizing networks as carriers to integrate computing, storage, data, and applications into a virtualized distributed computing platform. It is regarded as a new type of scalable, highly available, efficient, high-quality, and reliable computing platform.

2. Optimized Ant Colony Optimization Algorithm

2.1 Ant Colony Algorithm and Ant System

The ant colony algorithm has the following characteristics: (1) The pheromone concentration on each node of each route increases proportionally. Each ant calculates the probability of encountering each node on its route based on the surrounding pheromone concentration and selects the one with the highest probability as the next target. (2) To prevent the ant colony from entering into a program's "dead loop," the visited nodes in the loop are marked and prohibited from being visited again. (3) After completing a route, the ant adds corresponding pheromone to it based on the length of the route. If there are few ants on this route, the accumulated pheromone will decrease over time, indicating a low probability of success, and fewer ants will choose this route in the future.

2.2 Improved Ant System

Map/Reduce, as a widely adopted large-scale data processing program, divides a large number of user tasks into several smaller sub-tasks. By utilizing the virtual resource allocation mechanism, idle node resources in the network are allocated to these sub-tasks. In cloud computing, although each node can execute multiple tasks, assigning user tasks to the most efficient nodes as much as possible greatly improves the overall performance of the cloud computing system. Based on this, we set an upper time limit for task completion on each node. As long as this limit is not exceeded, the node is considered valid and can be assigned to the next handover point.

2.2.1 Information Pheromone

According to the number of CPUs and their processing power, as well as the capacity of memory, storage, and bandwidth, the information pheromones of virtual nodes are compared. The initialization process for the information pheromones of CPU computing power, memory, storage, and bandwidth is as follows:

$$\text{CPU information pheromone} = \frac{\text{Quantity} * \text{single threshold computing power}}{\text{CPU computing power threshold}}$$

$$\text{Memory information pheromone} = \frac{\text{Memory capacity}}{\text{Memory capacity threshold}}$$

$$Storage\ information\ pheromone\ = \frac{\text{Storage capacity}}{\text{Storage capacity threshold}}$$

$$\text{Bandwidth information pheromone} = \frac{\text{Bandwidth capacity}}{\text{Bandwidth capacity threshold}}$$

The information pheromone on each virtual machine node is the weighted sum of the individual hardware information pheromones. When the importance of hardware differs, the weight coefficients also differ. Specifically, the node information pheromone is calculated as follows: node information pheromone = w1 * CPU information pheromone + w2 * memory information pheromone + w3 * storage information pheromone + w4 * bandwidth information pheromone. In the process of executing the algorithm w1 + w2 + w3 + w4 = 1, the information pheromones need to be dynamically adjusted. When a valid node is assigned to a new task, the CPU on that node will have a higher utilization rate, and the information pheromone on that node will be correspondingly reduced.

2.2.2 Virtual Machine Resource Scheduling Algorithm

In this project, we plan to optimize the ant colony system in cloud computing from the following aspects: (1) Initialize the information pheromones of Worker nodes. (2) Submit user tasks containing multiple tasks to the master node for allocation. (3) The master node extracts the queued jobs in order of their quantity and allocates the corresponding ant colony on the nodes according to the quantity of jobs. If the tasks on this node are 'a', the master node will assign 'a * b' ants to explore the paths and start the exploration timer. (4) According to the preset rules of each ant, select the next route to take and mark the routes taken on the path. If it is valid, duplicate the route taken and return along the original path; otherwise, continue to evaluate and select higher-level nodes until a more efficient node is found, and repeat this process. (5) If an ant reports a valid node to the master node before the timer count ends, indicating the existence of a valid node, it will be assigned to the newly received user. (6) If a node is assigned to a user task or has completed a user task, the information pheromones on that node will be updated accordingly to indicate whether the node is occupied and available for use. (7) Repeat steps (3) and (6) until all tasks in the user job are fully allocated.

2.3 Improved Ant Colony Algorithm

To address this problem, this paper proposes an improved ant colony optimization method.

2.3.1 Ant Encounter Mechanism

Based on the movement direction of ants in the node network, they are distinguished into two types: forward ants and backward ants. The former is used to search for valid virtual resource nodes in the cloud computing environment. Once a valid virtual machine node is identified, the backward ant reports it to the master node. During the return journey, the backward ant marks the information pheromones of all available resources. In the initial stage, the system assigns a storage space to each node to store the node information attached to the backward ant, and sets an account timer. If a forward ant arrives at this node before the timer expires, it is considered an encounter between two ants at this node. When the timer reaches zero, the system automatically cleans up the information recorded on the node to prepare for updating with possible new information. The same node may be accessed by more than one backward ant at different times, and these backward ants come from multiple valid nodes that have been found, so the algorithm needs to distinguish which backward ant is generated by which valid node.

2.3.2 Forward Ant Movement Rules

When a forward ant moves to the next node, it will encounter two situations: either the forward ant has not encountered any backward ant yet, or the forward ant encounters a backward ant while searching for the next movement node. In the latter case, in order to improve the efficiency of finding the next movement node, it is necessary to refer to the information provided by the backward ant. Based on this, the forward ant colony should choose the appropriate forward node according to different situations. When the forward ant has not encountered a backward ant, the probability of each node appearing is calculated using the following formula, and the node with the highest probability is selected as the direction of advancement.

$$\text{The probability from node i to node j} = \frac{\text{The information pheromone of node j}}{\text{The expected time of node j} * \sum_{\text{The optional traversal node m}} \text{The information pheromone of node m / The expected time of node m}}$$

If the forward ant encounters a reflected ant at a node, two scenarios must be considered before selecting the next forward node: (1) If the forward ant and the reflected ant meet at a certain node, and this node only indicates the information carried by a single reflected ant, while the adjacent nodes also mark information from the reflected ant, then the probabilities of selecting each candidate node need to be calculated and compared with the probabilities of the adjacent nodes. Based on this, the optimal choice is made. (2) If the forward and backward ants meet at a node, and the encountered node has already registered information from multiple backward ants, it is necessary to select the node with the highest probability of effectiveness from the group of nodes marked by all reflected ants and the group of candidate nodes as the next movement node.

3. Experimental Results and Analysis

In the experiments, the information pheromone and expected time are represented by α and β, respectively, and the number of ants is represented by n. Based on this, we further study the optimal combination of α, β, and n. The scaling coefficients mentioned in the paper are all set to 0.2. In this experiment, 150 virtual machine nodes were set up, each with around 500 MIPS of dual-core computing power. The memory size was 200M, the bandwidth was 2M/s, and the external memory was 1G. On this basis, the project divided a task into 20 subtasks and repeated the task 10 times, and then used the improved ant colony algorithm and the improved ant colony algorithm for optimal allocation of the task in the cloud computing center. Before conducting the experiment, it is necessary to obtain the best combination of α, β, and n, which can be obtained through repeated trials (see Table 1).

Table 1 Experimental data

Experiment	α	β	n	Time
1	1	1	1.5	4631
2	1	2	1.5	4628
3	2	1	1.5	4605
4	1	1	2	4603
5	1	2	2	4602
6	2	1	2	4571
7	1	1	2.5	4585
8	1	2	2.5	4508
9	2	1	2.5	4553
10	1	1	3	4625

According to the experimental data in Table 1, the combination of α=1, β=2, and n=2.5 showed the best performance. Using this combination as the parameters, experiments were conducted on a workload of 20 tasks and improvements were made (refer to Table 2).

Table 2 Experimental results

Algorithm	α	β	n	Time
ACO1	1	2	2.5	4432
ACO2	1	2	2.5	4405
Improved Ant Model	1	2	2.5	4508

4. Conclusion

In conclusion, by incorporating the ant encounter mechanism and utilizing the reverse ant labeling of efficient resource nodes, the efficiency of forward ants in searching virtual machine resources can be improved. Therefore, in terms of task allocation time, using the improved ant colony algorithm is superior to using the improved ant system model.

References

[1] Pan, L. (2021). A high-performance embedded cloud computing platform architecture. Communications Technology, 54(05), 1184-1188.

[2] Guan, K. (2016). High-performance computing in OpenStack cloud platform using GPU passthrough. Computers and Information Technology, 24(06), 11-13.

About author: Xionggang Yang(1979-08), male, professional title: senior lecturer, education: undergraduate, research direction: virtualization technology.