Research of Ant Colony Algorithm Based on Multi-objective Optimization in Cloud Platform Virtual Machine Initialization

Zhen Wang\textsuperscript{1,a} Jing Du\textsuperscript{2,b} Quanqiang Miao\textsuperscript{3,c} Ruosi Cheng\textsuperscript{4,d}

\textsuperscript{1}Luoyang Electronic Equipment Testing Center, Henan, China
\textsuperscript{2}Luoyang Electronic Equipment Testing Center, Henan, China
\textsuperscript{3}Luoyang Electronic Equipment Testing Center, Henan, China
\textsuperscript{4}Luoyang Electronic Equipment Testing Center, Henan, China
\textsuperscript{a}zlgglz1314@foxmail.com \textsuperscript{b}jdstarry@aliyun.com \textsuperscript{c}miaquanqiang@163.com \textsuperscript{d}lincoln_cheng@foxmail.com

ABSTRACT: In order to solve the problem of resources waste in the process of virtual machine deployment in the cloud platform, we propose an ant colony algorithm based on multi-objective optimization. Firstly, the multi-objective optimization combination model is established. Then, we apply the model to ant colony algorithm. Finally, we modify the ant colony algorithm to adapt to cloud platform virtual machine initialization. In order to verify the effectiveness of the algorithm, cloudsim cloud simulation platform was used to carry out simulation experiments. The experimental results showed that the algorithm proposed in this paper could reach a relatively good level in SLA violation rate, power consumption, and resource loss of cloud platform compared with other commonly used algorithms.

CCS Concepts

• Networks→Network services→Cloud Computing.

1. INTRODUCTION

Although the cloud platform model is much better than the traditional single model, it still needs to consume a lot of power resources. Therefore, how to deploy and manage cloud platform resources to improve the utilization rate of resources has become a key issue that needs to be studied. In addition, resources should also be reasonably allocated to ensure cloud platform service quality and load balance.

Virtual machine initialization placement in cloud platform refers to how to place several virtual machines in a data center or multiple physical servers in the initial deployment stage of cloud platform. In the past, heuristic method or genetic algorithm is used to find the optimal placement solution in the research of virtual machine initialization placement. Because heuristic method does not have the ability of global optimization, it often falls into the local optimal solution. Genetic algorithm has a good search ability of global, but it is difficult to use feedback information. In this paper, multi-
objective optimization problem is applied to virtual machine placement. And an ant colony algorithm based on multi-objective optimization is proposed. As a result of the characteristics of ant colony algorithm, this algorithm can converge to the optimal solution efficiently.

2. RELATED WORK

Some scholars have also studied the problem of virtual machine initialization in the cloud platform. For example, literature[1] proposed an improved descending order adaptive algorithm for the placement of virtual machines for the first time. Literature[2] proposed a new method named EnaCloud, which can achieve dynamic placement of virtual machines, real-time optimization of virtual machine placement, and ultimately achieve the goal of high efficiency and energy saving. Literature[3] proposed multi-objective optimization algorithms based on non-dominated Sorting Genetic Algorithm, but they did not make full use of the feedback information of the system, making the solution less accurate. Literature[4] describes the node integration of virtual data center as a random packing optimization problem to deal with the randomness of load, which only considers CPU resources and does not treat other resources as a constraint condition. Literature[5][6] describe the dynamic migration of virtual machines as an optimization problem, and the optimization goal is to minimize energy consumption. At present, most virtual machine placement optimization methods are to transform multi-objective optimization problems into several single-objective optimization problems to be solved in stages. It is rare to optimize multiple objectives at the same time. Meanwhile, the results obtained are mostly local optimal rather than global optimal.

3. ANT COLONY ALGORITHM BASED ON MULTI-OBJECTIVE OPTIMIZATION

In generally, the optimal solution of multi-objective optimization problem can be obtained through enumeration, but due to the huge number of virtual machines in the cloud platform, the number of feasible solutions will be a large set, so the traditional method is no longer applicable. In this paper, an ant colony algorithm is proposed to realize multi-objective optimization, and an ant colony algorithm for multi-objective optimization under the initialization placement of cloud platform virtual machine is proposed.

3.1 Fitness Function

The first step to implement ant colony algorithm is to define the appropriate fitness function[7]. On above, the shortest route to the food source is one of the ways in which more pheromones can be stored. The problem of virtual machine initialization placement can be evaluated by the size of the fitness function. Getting the larger the fitness function value, the solution obtained by this method will be better. Virtual machine initialization placement requires three conditions to be considered and a fitness function for each condition is established, as described below:

(1) SLA violation rate function

\[ f_{SLA}(u_{CPU}) = \frac{1}{1 + e^{u_{CPU} - 0.5}} \]  

In order to reduce SLA violation rates, this chapter specifies that CPU utilization should not exceed 90%. The value range of \( f_{SLA}(u_{CPU}) \) is between [0-1].

(2) Resource utilization function

\[ f_{resource}(u_{CPU}, u_{mem}, u_{bw}) = u_{CPU} \times u_{mem} \times u_{bw} \]  

According to the formula, the utilization rate of the overall resource is proportional to the utilization rate of each resource. The value range of \( f_{resource} \) is between [0-1].

(3) Power consumption function
\[ f_{power}(u_{CPU}) = \frac{u_{CPU}}{P_{idle} + (P_{busy} - P_{idle}) \times u_{CPU}} \times P_{busy} \]  \hspace{1cm} (3-3)

\( f_{power} \) -- power consumption rate of the host;

\( u_{CPU} \) -- CPU utilization on the host;

\( P_{idle} \) -- power consumption in idle load of host CPU;

\( P_{busy} \) -- power consumption at full load of the host CPU.

For the sake of convenience, the paper only reflects the change of power consumption according to the use of CPU. The value range of \( f_{power} \) is between [0-1].

Combined with the formulas (3-1,3-2,3-3), the following is the multi-objective optimization formula based on weighting coefficient\[^{[8]}\], which is taken as the fitness function of the multi-objective optimization ant colony algorithm in the virtual machine initialization placement. The definition of the formula is shown as follows:

\[ f(u_{CPU}, u_{mem}, u_{bw}) = k_1 f_{SLA}(u_{CPU}) + k_2 f_{resource}(u_{CPU}, u_{mem}, u_{bw}) + k_3 f_{power}(u_{CPU}) \]  \hspace{1cm} (3-4)

Where, the coefficients \( k_1, k_2, k_3 > 0 \) are determined according to the importance of each target in the fitness function. In this paper, the three targets that need to be optimized are considered to have the same status, so the parameters of all three can be set to 1, namely \( k_1=1, k_2=1, k_3=1 \).

### 3.2 Pheromone Setting

The maximum-min-ant-system (MMAS)\[^{[9]}\] in ant colony algorithm is adopted to realize the virtual machine placement strategy of multi-objective optimization. The pheromone update rule in MMAS is that, after completing each cycle, only pheromones on the optimal solution are enhanced. At initialization time, the initial value of pheromone is the same between any host and virtual machine. For specific pheromone update rules, see formula (3-5):

\[ y_{iu}(n) = (1 - \rho) \times y_{iu}(n - 1) + \Delta y_{iu}^{best} \]  \hspace{1cm} (3-5)

\( \Delta y_{iu}^{best} = f(S_{best}) \) If the virtual machine i is loaded on node u

\( \rho \) -- the volatile coefficient of pheromone;

\( y_{iu}(n) \) -- after the n times iteration, the pheromone concentration of virtual machine i and corresponding host u;

\( y_{iu}(n - 1) \) -- after the n-1 times iteration, the pheromone concentration of virtual machine i and corresponding host u;

\( \Delta y_{iu}^{best} \) -- the loop optimal solution in the matching between the virtual machine and host pheromone increment;

\( f(S_{best}) \) -- pheromone increment function, which can be represented by fitness function, where \( S_{best} \) represents the optimal solution set.

In order to avoid the infinite increase of information value, \( \rho \) range between [0-1]. In MMAS, pheromones are updated only on the optimal solution after completing each cycle. This can easily cause the search to stop and cause subsequent ants to choose the same path. To prevent this from happening, MMAS limits the values of pheromones. We limit \( y_{iu} \) to the interval \( [y_{min}, y_{max}] \). Here, \( y_{max} \) is set to the initial value of \( y_{iu} \), and \( y_{min} \) is set to \( y_{min} = y_{max} / g \) (\( g > 1 \)). Specify that when the value of \( y_{iu} \) is greater than \( y_{max} \), set \( y_{iu} = y_{max} \); When the value of \( y_{iu} \) is less than \( y_{min} \), set \( y_{iu} = y_{min} \).

### 3.3 Transfer Probability Function

Although ant colony algorithm has strong search ability, it has the disadvantage of slow search speed compared with other algorithms. Ant colony algorithm, transfer probability function is an important index of its convergence. Reasonable design of transfer probability function can accelerate the search speed of the algorithm. Therefore, a taboo list is designed to make it easier and faster for ants to search in the virtual machine. This taboo list records the number of virtual machines that an ant knows cannot load and the number of virtual machines that the ant has searched for before searching host u for a
virtual machine that can be placed. It is important to note that after completing one loop, the tabulation table is cleared so that the ants can select all available virtual machines on the next loop.

In this chapter, the transition probability function is defined as shown in formula (3-6):

\[
P^{k}_{iu}(t) = \begin{cases} 
\frac{\gamma^{k}_{iu}(t) \times \mu^{k}_{iu}(t)}{\sum_{i \in \text{allowed}_k} \gamma^{k}_{iu}(t) \times \mu^{k}_{iu}(t)} & i \in \text{allowed}_k \\
0 & \text{others}
\end{cases}
\]  

(3-6)

\[P^{k}_{iu}(t)\] -- the transfer probability of ant \(k\);

\[\text{allowed}_k\] -- the next set of virtual machines that ant \(k\) can choose is the set of virtual machines that can be assigned after removing the set of virtual machines in the taboo list;

\[\alpha\] -- the relative importance of residual information. When \(\alpha = 0\), the algorithm is the traditional greedy algorithm;

\[\beta\] -- the relative importance of predicted values. When \(\beta = 0\), the algorithm becomes a heuristic algorithm of positive feedback;

\[\gamma^{k}_{iu}(t)\] -- the pheromone intensity of virtual machine \(i\) to host \(u\);

\[\mu^{k}_{iu}(t)\] -- visibility factor, which represents the probability that ants search virtual machine \(i\), the value varies with virtual machine, the specific expression is:

\[
\mu^{k}_{iu}(t) = \frac{1}{1 - r^{\text{CPU}}_i} \times \frac{1}{1 - r^{\text{mem}}_i} \times \frac{1}{1 - r^{\text{bw}}_i}
\]  

(3-7)

\[r^{\text{CPU}}_i\] -- virtual machine \(i\) request the CPU resources of the host \(u\) on the proportion of remaining CPU resources;

\[r^{\text{mem}}_i\] -- virtual machine \(i\) request memory resources accounted for the proportion of the remaining memory resources on host \(u\);

\[r^{\text{bw}}_i\] -- virtual machine \(i\) request network bandwidth resources accounted for the proportion of host \(u\) on the rest of the network bandwidth resources.

According to the formula (3-7), the size of \(\mu^{k}_{iu}(t)\) is proportional to the resources that the virtual machine has, that is, the virtual machine has a high visibility factor, the virtual machine will be the priority to choose.

### 3.4 Construction of The Optimal Solution Set

The optimal solution of the multi-objective problem is a set of optimal solutions. Our goal is to find a solution in the optimal solution set as close as possible to the Pareto optimal boundary. The method of comparison and exclusion is used to realize the construction process of Pareto solution set\(^{[10]}\). Roughly, the processing process is divided into two steps. The first step is that in the first cycle, each ant liberates the local optimal solution found by itself to an optimal solution set of this cycle. The second step is to compare the solution set of the optimal solution of this cycle generated in the first step after the end of this cycle, and then select the required optimal solution according to the corresponding rules.

The calculation comparison method to construct the Pareto optimal solution set of detailed steps as follows: Getting an optimal solution sets of one cycle \(S^{*}_{\text{cycle}} = \{s_1, s_2, ..., s_k\}\), \(k\) represents the number of local optimal solutions searched by ants. According to the formula (3-4), each solution in this chapter has three components, which correspond to each objective to be optimized. Therefore, the sub-goals of each solution are evaluated separately at first, which is achieved by dominating the solution set\(^{[11]}\). Compare the rest of the solutions in \(S^{*}_{\text{cycle}}\) in pairs. An optimal solution in this cycle, \(S^{*}_{\text{cycle}}\), is obtained. After obtaining the optimal solution of this cycle, the new global optimal solution \(S^{\text{best}}\) was obtained by comparing it with the old global optimal solution \(S^{\text{best}}\) according to the way of comparison and exclusion. When all the loop searches are completed, the optimal solution set of Pareto we seek is the global optimal solution set \(S^{\text{best}}\).

### 4. SIMULATION RESULTS

#### 4.1 Establishment of Simulation Experiment
This paper adopts CloudSim platform to simulate the initialization placement strategy of virtual machine\cite{12}. On the simulation, Best Fit Algorithm (BFA)\cite{13}, Improved Group Genetic Algorithm (IGGA)\cite{14}, single-objective Ant Colony Optimization algorithm (SACO) and the multi-objective Ant Colony Optimization algorithm (MACO) proposed in this paper are used for experiments. According to the experimental results, the merits and demerits of the algorithm in the virtual machine initialization placement are determined.

During the simulation of CloudSim platform, the number of hosts was set to 100, among which, the computing power of 30 CPU was 1000MIPS, the computing power of 40 CPU was 2000MIPS, and the computing power of 30 CPU was 3000MIPS. The size of memory, hard disk, network bandwidth of each host is the same, here the memory size is 10GB, the size of the hard disk is 2TB, network bandwidth is 1Gbps. Set the number of virtual machines that need to be initialized to be placed as 300, among which 120 CPU have the computing power of 250MIPS, 90 CPU the computing power of 500MIPS, 60 CPU the computing power of 750MIPS, and 30 CPU have the computing power of 1000MIPS. Each virtual machine has the same size of memory, hard disk and network bandwidth, which are specified as 2GB memory size, 200GB hard disk size and 250Mbps network bandwidth. The power consumption of the host is $P_{idle}=175W$ when CPU utilization is 0%, and $P_{busy}=250W$ when CPU utilization is 100%.

4.2 Simulation Results and Analysis

Figure 4-1 shows the comparison of SLA violation rate of five virtual machine initialization placement methods: BFA-CPU, IGGA, S-SACO, P-SACO and MACO. Since the memory and broadband resources of virtual machines in simulation are the same, the optimal adaptive algorithm BFA only selects the optimal adaptive algorithm BFA-CPU for CPU.

![Figure 4-1 Comparison of SLA violation rates of the five algorithms](image.png)

The specific comparative analysis is as follows: according to figure 4-1, SLA violation rate of the two virtual machine placement policies, namely, BFA-CPU and P-SACO, is the highest. The MACO algorithm proposed in this chapter can optimize multiple targets according to the actual situation, so as to achieve good optimization on a certain sub-target.

Figure 4-2 and figure 4-3 respectively show the comparison between resource waste and power consumption of five virtual machine initialization placement methods: BFA-CPU, IGGA, S-SACO, P-SACO and MACO. As can be seen from the comparison of the two figures, the comparison between resource waste and power consumption in the five placement methods is the same. This is because the waste of resources will lead to the increase of electric energy consumption, so the waste of resources and the consumption of electric energy can be seen as a proportional distribution to a certain extent, so the unified analysis of the waste of resources and the consumption of electric energy will be done here.
Resources waste and the power consumption of the highest is the S-SACO. Resources waste and the power consumption of the lowest is the P-SACO, BFA-CPU, IGGA, MACO of resource waste rate and power consumption at an intermediate level. MACO proposed in this chapter, both the global search ability, and have the characteristics of multi-objective optimization, so the waste of resources and power consumption performance is better.

In conclusion, the solution sets of S-SACO and P-SACO than those of the other three when only one optimization goal needs to be considered. Compared with the other four algorithms, the MACO algorithm proposed in this chapter can obtain the optimal solution set when the virtual machine initialization placement requires multiple objective optimization.

5. CONCLUSION
This paper mainly focuses on the problem of the initialization and placement of virtual machine on cloud platform. Firstly, the requirements of the virtual machine initialization are analyzed, then the research status of virtual machine initialization is summarized. Thirdly, the corresponding solution is proposed. Finally, experiments are carried out to verify the advantages of the proposed algorithm in SLA violation rate, power consumption and resource utilization.

The initialization placement strategy of virtual machine proposed by CloudSim cloud simulation platform is simulated. According to the results, it can be seen that the scheduling strategy proposed in this paper can well balance the relationship among SLA violation rate, power consumption and
resource utilization, so that the performance of each factor of virtual machine is relatively better. Although it is not as good as the single-objective optimization method specifically targeted in the optimization of a certain target, compared with the traditional heuristic method and genetic algorithm, the proposed algorithm can not only well solve the problem of multiple conflicting goals, but also make SLA violation probability, resource waste and power consumption reach a very low level.

REFERENCES

[1] Verma A, Ahuja P, Neogi A. pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems[M]// Issarny V, Schantz R. Lecture Notes in Computer Science. 2008: 243-264.

[2] Li B, Li J, Huai J, et al. EnaCloud: An Energy-saving Application Live Placement Approach for Cloud Computing Environments[M]. IEEE International Conference on Cloud Computing. 2009: 17-24.

[3] Li Q, Hao Q, Xiao L, et al. Adaptive Management and Multi-Objective Optimization for Virtual Machine Placement in Cloud Computing[J]. Chinese Journal of Computers. 2011, 2253-2264.

[4] Chen M, Zhang H, Su Y Y, et al. Effective VM sizing in virtualized data centers[C]// Ifip/ieee International Symposium on Integrated Network Management. IEEE, 2011:594-601.

[5] Wang X, Wang Y. Coordinating Power Control and Performance Management for Virtualized Server Clusters[J]. IEEE Transactions on Parallel & Distributed Systems, 2010, 22(2):245-259.

[6] Jung G, Hiltunen M A, Joshi K R, et al. Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures[C]// IEEE, International Conference on Distributed Computing Systems. IEEE, 2010:62-73.

[7] Li Lirong, Zheng Jinhua. Multi-objective genetic algorithm based on Pareto Front [J]. Journal of Natural Science, Xiangtan University. 2004 (01): 39-41.

[8] Xiao Xiaowei, Xiao Di, Lin Jinguo et al. Overview of the research on multi-objective optimization problems [J]. Computer Applied Research. 2011 (03): 805-808.

[9] Le D N, Bhateja V, Nguyen G N. A parallel max-min ant system algorithm for dynamic resource allocation to support QoS requirements[C]// IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics. IEEE, 2017:697-700.

[10] Luo Yong, Guo Yamo, Liu Chong. Fuzzy classification system design based on Pareto fireworks algorithm [J]. Computer Engineering, 2017, 43 (2): 304-307.

[11] Nederlof J, Rooij J M M V. Inclusion/Exclusion Branching for Partial Dominating Set and Set Splitting[C]// Parameterized and Exact Computation, International Symposium, Ipec 2010, Chennai, India, December 13-15, 2010. Proceedings. DBLP, 2010:204-215.

[12] Shah J, Malik LG. Digital Forensic in Cloudsim[M]// Jain LC, Behera HS, Mandal JK, et al. Smart Innovation Systems and Technologies. 2015: 563-572.

[13] Varsamis D, Chanioglou F. A Parallel Approach of Best Fit Decreasing Algorithm[J]. Wseas Transactions on Computers, 2018, 17(9):79-85.

[14] Li Jinchao, Chen Jingyi, Wu Jie and others. Research on virtual machine placement based on improved grouping genetic algorithm [J]. Computer engineering and design. 2012 (05): 2053-2056.