Postprint of Virtual Machine Placement Method Based on Improved Ant Lion Algorithm

Authors: Liu Yaohong, Wang Yong, Wang Yong

Date: 2022-11-02T00:00:00+00:00

Abstract

Virtual machine placement is a critical step in the virtual machine consolidation process, and the effectiveness of virtual machine placement methods often impacts the resource utilization efficiency and performance of cloud data centers. Such problems can be addressed by establishing multi-objective optimization models. Current cloud data centers exhibit high energy consumption, low resource utilization, and resource fragmentation. To tackle these issues, we propose a virtual machine placement strategy based on the MALO algorithm. By constructing a multi-objective, multi-constraint virtual machine placement model, we optimize three aspects: energy consumption, resource utilization, and resource fragmentation. Building upon the ant lion algorithm, we enhance the solution space boundary variation strategy and the ant random walk position selection strategy, and finally correct for out-of-bounds ant positions, thereby better ensuring population diversity and enabling more effective escape from local optima. Simulation experiments conducted on a virtual machine placement platform compare the MALO algorithm with four other virtual machine placement algorithms. Experimental results demonstrate that, compared with the ant lion algorithm, BRC algorithm, MBFD algorithm, and FFD algorithm, the MALO algorithm achieves certain improvements in reducing energy consumption, improving resource utilization, and decreasing resource fragmentation.

Full Text

Virtual Machine Placement Method Based on Improved Ant Lion Optimizer

Yaohong Liu[1,2] [1]School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi, China
[2]Guangxi Key Laboratory of Cloud Computing and Complex Systems, Guilin University of Electronic Technology, Guilin, Guangxi, China

Abstract

Virtual machine placement is a critical step in the virtual machine consolidation process. Such problems can be solved by establishing multi-objective optimization models. The quality of virtual machine placement methods often affects the resource utilization efficiency and performance of cloud data centers. Current cloud data centers suffer from high energy consumption, low resource utilization, and resource fragmentation. To address these issues, this paper proposes a virtual machine placement strategy based on an improved ant lion optimizer. By establishing a multi-objective, multi-constrained virtual machine placement model, optimization is performed across multiple dimensions. Based on the ant lion algorithm, improvements are made to the boundary variation strategy of the solution space and the ant' s random walk position selection strategy. Finally, corrections are applied when ant positions exceed boundaries, ensuring better population diversity and enabling more effective escape from local optima. The proposed algorithm is compared with several other virtual machine placement algorithms through simulation experiments.

Keywords: virtual machine placement; cloud data center; multi-objective optimization; ant lion algorithm; energy consumption

1. Introduction

Virtual machine placement is a key step in the virtual machine consolidation process. The quality of placement strategies directly impacts resource utilization efficiency and overall performance in cloud data centers. Currently, cloud data centers face significant challenges including high energy consumption, low resource utilization rates, and resource fragmentation. Virtualization technology has emerged as a focal point for addressing these issues, enabling the integration of software and hardware resources to meet user computing demands. However, energy management remains a critical concern, as idle physical machines consume approximately 70% of their peak power, contributing substantially to overall data center energy consumption and associated environmental costs.

Existing research has employed various intelligent optimization algorithms to solve the virtual machine placement problem. Parvizi et al. [12] addressed energy consumption, resource utilization, and the number of active physical machines using the NSGA-III genetic algorithm, demonstrating advantages over baseline algorithms. Li Kaiqing et al. [13] proposed a discrete bat algorithm for virtual machine placement (DBA-VMP), achieving improvements in reducing energy consumption and improving resource utilization with relatively low computational overhead. Ma Xiaojin et al. [15] developed an improved simulated annealing algorithm for virtual machine scheduling, introducing parallel concepts to avoid local optima. Tang et al. [16] proposed a hybrid genetic algorithm considering both physical machine and network communication energy consumption, showing higher performance and efficiency compared to standard genetic algorithms.

However, these studies have limitations. Most heuristic approaches focus solely on energy consumption, resource utilization, and load balancing while neglecting resource fragmentation, which is also a critical metric affecting consolidation effectiveness. Furthermore, approximation algorithms that consider only current load conditions are prone to falling into local optima. To overcome these shortcomings, this paper proposes a virtual machine placement algorithm based on an improved ant lion optimizer (MALO), which enhances population diversity and randomness to effectively avoid local optima.

---

2. Virtual Machine Placement Problem Definition

The virtual machine placement problem can be essentially modeled as a multi-dimensional bin packing problem. Consider a cloud data center with m virtual machines and n physical machines. The problem involves finding a reasonable scheduling strategy to place m virtual machines onto n physical machines while satisfying specific constraints.

Let the virtual machine list be $VM = \{vm_1, vm_2, ..., vm_m\}$ and the physical machine list be $PM = \{pm_1, pm_2, ..., pm_n\}$. Each virtual machine vm_i requests resources $R_i = (u_i, w_i)$, where u_i represents CPU resources and w_i represents memory resources requested by the i-th virtual machine. The resource capacity of physical machines is defined as $R_{pm} = \{(c_j, m_j) | j = 1, 2, ..., n\}$, where c_j and m_j denote the CPU and memory capacities of the j-th physical machine, respectively.

The placement must satisfy the following constraints:

1. Each virtual machine can be placed on at most one physical machine: $\sum_{j=1}^{n} x_{ij} = 1$ for $i = 1, 2, ..., m$, where $x_{ij} \in \{0, 1\}$ indicates whether virtual machine vm_i is placed on physical machine pm_j.
2. The total resource requests of all virtual machines on any physical machine cannot exceed its capacity: $\sum_{i=1}^{m} x_{ij} u_i \leq c_j$ and $\sum_{i=1}^{m} x_{ij} w_i \leq m_j$ for $j = 1, 2, ..., n$.

---

3. Optimization Models

3.1 Energy Consumption Model Energy consumption is a primary concern in data centers. For a data center with m virtual machines and n physical machines, the power consumption of physical machine pm_i is modeled as:

$$P_i = hfull\_poweri \times (\alpha + (1 - \alpha) \times hcpu\_utili)$$

where $hfull\_poweri$ is the power consumption of pm_i at full load, α represents the percentage of idle power consumption relative to full load (typically 0.6-0.7, uniformly set to 0.7 in this study), and $hcpu\_utili$ is the CPU utilization of pm_i. The total energy consumption of the data center is:

$$P_{total} = \sum_{i=1}^{n} P_i \times y_i$$

where y_i indicates whether pm_i is active ($y_i = 1$) or inactive ($y_i = 0$).

3.2 Resource Utilization Model Resource utilization is a key metric for evaluating placement strategies. For m virtual machines and n active physical machines, the average resource utilization is calculated as:

$$hcpu\_avg = \frac{\sum_{j=1}^{n} hcpu\_usedj}{\sum_{j=1}^{n} hcpu\_maxj}, \quad hmem\_avg = \frac{\sum_{j=1}^{n} hmem\_usedj}{\sum_{j=1}^{n} hmem\_maxj}$$

where $hcpu\_usedj = \sum_{i=1}^{m} u_i x_{ij}$ and $hmem\_usedj = \sum_{i=1}^{m} w_i x_{ij}$ represent the total CPU and memory resources requested by virtual machines on pm_j, and $hcpu\_maxj$ and $hmem\_maxj$ denote the total CPU and memory capacities of pm_j.

3.3 Resource Balance Model To achieve balanced placement, we propose a ratio-based resource balance model. For physical machine pm_i, the balance is measured by how close the ratio of CPU utilization to memory utilization is to 1:

$$Balance_i = \left| \frac{hcpu\_utili}{hmem\_utili} - 1 \right|$$

The overall resource balance degree of the data center is:

$$Balance_{total} = \sum_{i=1}^{n} Balance_i \times y_i$$

3.4 Resource Fragmentation Model Unreasonable virtual machine placement leads to resource fragmentation, where remaining resources on a physical machine cannot satisfy new virtual machine requests. Resource fragmentation is defined as:

$$f_{cpu} = \sum_{i=1}^{n} hcpu\_remi \times y_i, \quad f_{mem} = \sum_{i=1}^{n} hmem\_remi \times y_i$$

where $hcpu\_remi$ and $hmem\_remi$ represent the remaining CPU and memory resources on pm_i. A physical machine is considered to have resource fragments when its remaining CPU $\leq$ 512 MB or remaining memory $\leq$ 512 MB.

---

4. Objective Function

The optimization objectives are to minimize energy consumption, maximize resource utilization, and reduce resource fragmentation. The comprehensive objective function is:

$$F = \min(L + r \times E)$$

where L represents the normalized resource balance degree, E represents normalized energy consumption, and r is a weighting parameter. When r is larger, energy reduction is prioritized; when r is smaller, resource balance and fragmentation reduction are emphasized. To balance these objectives, r is set to 0.5 in this study.

---

5. Improved Ant Lion Optimizer (MALO)

The standard ant lion optimizer (ALO) tends to converge prematurely and struggle with escaping local optima. To address this, we propose MALO with two key improvements:

5.1 Adaptive Boundary Adjustment Mechanism In traditional ALO, the boundary contraction during ant random walks is linear, which is unsuitable for cloud environment encoding and may destroy solution diversity. MALO introduces an adaptive boundary adjustment mechanism:

$$I = 10^W \times \frac{t}{T}$$

where W is dynamically adjusted based on iteration count, t is the current iteration, and T is the total number of iterations. This non-linear boundary change better preserves population diversity.

5.2 Ant Position Update Strategy In standard ALO, ants walk randomly around antlions selected by roulette wheel and the elite antlion, with the final position being the midpoint of these two positions. This homogenizes ant movement trends and destroys population diversity. MALO improves this by biasing the random walk differently across iterations:

$$\mathrm{Pos}_{Ant}^{t+1} = \left(1 - \frac{t}{T}\right) \times \mathrm{Pos}_{Roulette}^{t} + \frac{t}{T} \times \mathrm{Pos}_{Elite}^{t} + \mathrm{rand} \times (\mathrm{Pos}_{Elite}^{t} - \mathrm{Pos}_{Roulette}^{t})$$

where $\mathrm{Pos}_{Roulette}^{t}$ is the position of the roulette-wheel selected antlion, Pos_{Elite}^{t} is the elite antlion position, and Pos_{Ant}^{t+1} is the updated ant position. Early iterations bias toward the roulette-wheel antlion, while later iterations bias toward the elite antlion, ensuring both exploration and exploitation.

5.3 Boundary Correction Strategy When ants exceed boundaries, traditional ALO moves them directly to the boundary, harming diversity. MALO instead applies random correction:

$$\text{if } \mathrm{Pos}_{Ant} > bu \text{ or } \mathrm{Pos}_{Ant} < bl: \quad \mathrm{Pos}_{Ant} = bl + \mathrm{rand} \times (bu - bl)$$

where bu and bl are the upper and lower bounds, and rand is a random number in $(0, 1)$. This preserves population diversity.

---

6. Experimental Setup

We developed a simulation platform in Python for virtual machine placement that can read, process, and statistically analyze data. The BitBrains dataset [20], from a real data center providing cloud services, was used for evaluation. The experimental scenario employs heterogeneous data center configurations with physical machine specifications shown in Table 1.

The data center contains physical machines with varying configurations. Experiments were conducted with different numbers of virtual machines to be placed: 50, 60, 70, 80, and 90. The population size for MALO and ALO was set to 50, with maximum iterations of 100. Other comparison algorithms (BRC, FFD, MBFD) used default parameters. Performance metrics include energy consumption, number of active physical machines, CPU utilization, memory utilization, and resource fragmentation.

---

7. Experimental Results and Analysis

7.1 Energy Consumption Figure 1 compares the energy consumption of MALO, ALO, BRC, FFD, and MBFD. MALO demonstrates consistent energy advantages across all VM scales. When the number of virtual machines is 50, MALO reduces energy consumption by 7.2% compared to standard ALO. The improvement is attributed to MALO' s better mechanism for escaping local optima, enabling more thorough searches and reducing the number of active physical machines.

[Figure 1: see original paper]

7.2 Active Physical Machines The number of active physical machines indirectly reflects energy consumption. Figure 2 shows that MALO maintains the fewest active physical machines across most scales. For 64 virtual machines, MALO and ALO show similar performance, but MALO outperforms others by 5-10% at scales of 50-90 VMs. This is crucial for addressing high energy consumption, as fewer active machines directly translate to lower energy usage.

[Figure 2: see original paper]

7.3 Resource Utilization Resource utilization is a key indicator of placement quality. Figures 3 and 4 show CPU and memory utilization, respectively. MALO achieves the highest CPU utilization across all scales, with advantages becoming more pronounced as VM numbers increase. For memory utilization, MALO performs best at scales of 60-90 VMs, though slightly lower than BRC at 50 VMs. Overall, MALO's ability to avoid local optima leads to more reasonable placement strategies and higher resource utilization.

[Figure 3: see original paper] [Figure 4: see original paper]

7.4 Resource Fragmentation Resource fragmentation measures placement balance. Figure 5 shows that MALO generally produces the least CPU fragmentation, second only to BRC. For memory fragmentation (Figure 6), MALO shows the most significant improvement, reducing fragmentation by 15-20% compared to ALO. This is because MALO considers resource balance during placement, minimizing imbalances that lead to fragmentation.

[Figure 5: see original paper] [Figure 6: see original paper]

---

8. Conclusion and Future Work

This paper proposes a virtual machine placement method based on an improved ant lion optimizer (MALO). By introducing adaptive boundary adjustment mechanisms and improved ant position update strategies, MALO enhances population diversity and provides better escape from local optima. Experimental comparisons with ALO, BRC, FFD, and MBFD demonstrate that MALO effectively reduces energy consumption, improves resource utilization, and decreases resource fragmentation.

Future work will consider integrating strategies from other intelligent optimization algorithms such as ant colony optimization to further enhance global search capabilities. Additionally, the improved ant lion algorithm can be applied not only to virtual machine placement but also to virtual machine migration processes. Finally, deploying the algorithm in real environments will further validate its effectiveness in improving cloud data center efficiency.

References

[1] Liu X, Zhao Y, Dong Y, et al. I-Neat: an intelligent adaptive virtual machine consolidation framework[J]. Tsinghua Science and Technology, 2022, 27(1): 13-26.

[2] Masdari M, ValiKardan S, Shahi Z, et al. An efficient energy-aware virtual machine placement scheme for cloud data centers[J]. Journal of Network and Computer Applications, 2016, 66: 106-127.

[3] Li Q, Jiang Y. Cloud computing system: current status and future trends[J]. Third International Symposium on Electronic Commerce and Security, 2010: 332-336.

[4] Farahnakian F, Ashraf A, Pahikkala T, et al. Using ant colony system to consolidate VMs for green cloud computing[J]. IEEE Transactions on Services Computing, 2015, 8(2): 187-198.

[5] Arroba P, Ayala J L, Moya J M, et al. Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers[J]. Concurrency and Computation: Practice and Experience, 2016, 29(10): 494-495.

[6] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers[J]. Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.

[7] Beloglazov A, Buyya R. Energy efficient resource management in virtualized cloud data centers[C]//2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010: 826-831.

[8] Alsbatini G, Ulusoy H. Efficient virtual machine placement algorithms for cloud data centers[J]. Computer Systems Science and Engineering, 2020, 17(1): 29-50.

[9] Jangiti S, Jayaraman P P, Mitra K, et al. Resource ratio-based multi-objective virtual machine placement algorithm[J]. Journal of Systems and Software, 2020, 105: 824-837.

[10] Tarafdar A, Debnath M, Khatua S. Energy and quality of service-aware virtual machine consolidation in heterogeneous cloud data center[J]. The Journal of Supercomputing, 2020, 76(11): 9095-9126.

[11] Farahnakian F, Pahikkala T, Liljeberg P, et al. Utilization prediction aware VM consolidation approach for green cloud computing[C]//2015 IEEE 8th International Conference on Cloud Computing, 2015: 381-388.

[12] Parvizi E, Rezvani H. Utilization-aware energy-efficient virtual machine placement approach in cloud data centers using meta-heuristic algorithms[J]. Cluster Computing, 2020, 23(4): 1-23.

[13] Li K, Xu J, Zhao G, et al. Multi-objective optimization of virtual machine placement in cloud environment[J]. Computer Applications, 2019, 39(12): 3597-3605.

[14] Li S, Wang Y, Zhang S. Virtual machine placement method based on Memetic algorithm[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2020, 32(3): 356-367.

[15] Ma X, Li Y, Wang H, et al. Virtual machine scheduling optimization method based on improved simulated annealing algorithm[J]. Computer Applications, 2018, 39(8): 2256-2261.

[16] Tang L, Pan Z, Zheng Y. A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers[J]. Neural Processing Letters, 2014, 41(2): 211-221.

[17] Li K, Xu J, Zhao G, et al. Multi-objective optimization of virtual machine placement in cloud environment[J]. Journal of Systems and Software, 2020, 105: 824-837.

[18] Ghasemi A, Abolfazli S. Multi-objective load balancing algorithm for virtual machine placement in cloud data centers[J]. Computing, 2020, 102(9): 2049-2072.

[19] Ammar M, Luo J, Tang Q. Intra-balance virtual machine placement for effective reduction of energy consumption and SLA violation[J]. IEEE Access, 2019, 7: 72387-72402.

[20] Shen Q, Beek V, Iosup A. Statistical characterization of business-critical workloads hosted in cloud datacenters[C]//2015 IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2015: 465-474.

Note: Figure translations are in progress. See original paper for figures.