Hindawi
Scientific Programming
Volume 2021, Article ID 6784419, 9 pages
https://doi.org/10.1155/2021/6784419

Research Article

A Specific Risk Evaluation System for Live Virtual Machine Migration Based on the Uncertain Theory

Hang Zhou ,[1,2] Xinying Zhu,[2] and Jian Wang[1]

[1] *Hangzhou Innovation Institute of Beihang University, Hangzhou, Zhejiang 310051, China*
[2] *Department of Engineering Network, Zhoukou Normal University, Zhoukou, Henan 466001, China*

Correspondence should be addressed to Hang Zhou; profzhou@shu.edu.cn

Received 15 September 2021; Revised 24 October 2021; Accepted 18 November 2021; Published 13 December 2021

Academic Editor: Rahman Ali

Benefiting from the convenience of virtualization, virtual machine migration is generally utilized to fulfil optimization objectives in cloud/edge computing. However, live migration has certain risks and unapt decision may lead to side effects and performance degradation. Leveraging modified deep Q network, this paper provided an advanced risk evaluation system. Thorough formulation was given in this paper and a specific integration method was innovated based on uncertain theory. Series experiments were carried on computing cluster with OpenStack. The experimental results showed deep Q network for risk system was reliable while the uncertain approach was a proper way to deal with the risk integration.

1. Introduction

Cloud computing has become a consolidated computing paradigm, which allows users around the world to submit various computing requests. Benefiting from convenience of virtualization, live migration of virtual machine is widely used to fulfil optimization objectives such as energy conversation, load balance, and rapid response.

Although cloud users are ignorant of computing details, the optimization of virtual resources in cluster is not a trivia. For IDC (Internet Data Center) engineers or management systems, lots of complex factors need to be taken into consideration. For instance, the source physical machine (PM), the destination PM, the target virtual machine (VM), the resource status along the route path, and the migration opportunity, all these factors are related to the optimal solution for live VM migration. As the optimal problem becomes more complex, AI technology is now utilized to provide deep analysis and meticulous operation. Leveraging innovative machine learning skills, Google has significantly improved efficiency of IDC in terms of resource utilization and energy conservation. Now deep learning network and reinforcement learning is widely utilized by cloud providers such as Amazon AWS, Tencent Cloud, and MS Azure.

Extreme optimization always involves invisible risks. Live migration may also lead to side effects and performance degradation when migration is overused or unreasonable migration process is carried on. The performance of application running on the migrating VM would be affected especially during the beginning and downtime of migration process [1]. Performance degradation in VMs certainly appears with high resource utilization on PMs. Experiments by researchers Xu et al. [2] showed application running on VMs enduring serious performance degradation and variation; e.g., the loading time of Doom3 Game will increase 25 to 110 percentage with an Amazon EC2 enduring resource contention with other instances. Research [3–5] demonstrated that the execution duration and response time of applications were affected by live migration which leads to SLAV (Service Level Agreement Violation) and extra energy consumption.

For live VM migration, how to make rational decision (whether to migrate or not) is a crucial issue. Due to the high complexity of heterogeneous computing cluster, heuristic algorithms with fixed migration threshold for CPU/memory utilization are usually applied to deal with primary optimization goals (e.g., load balance, energy conservation) in live VM migration. Bionics algorithms are more refined;

nevertheless they cannot cope with the frequently evolving environment (the resource configuration, the workload, and the requirement/constraint are constantly changing) in cloud computing. Different from these approaches, focusing on the live migration decision issue, this paper provides a specific risk evaluation method based on deep Q network and the uncertain theory. The unique highlight of this paper is to investigate the migration decision issue from AI (reinforcement learning approach) perspective. In fact, general AI algorithm cannot be applied easily for the complex environment and huge optimization space.

The contribution of this paper mainly lies in the specific design in which reinforcement learning and uncertainty theory are combined with harmony. In this paper, a specific three-level DQN framework is innovated according to the live VM migration environment. In addition, the uncertain theory is utilized to provide risk integration for DQN system. As far as we know, this is the first attempt to combine uncertain theory and DQN structure in real distributed computing scenario.

The remainder of this paper is organized as follows. Related works are discussed in Section 2. Section 3 presents formal definition while we describe the details of innovative algorithm in Section 4. The corresponding performance evaluation is taken in Section 5 while conclusion and future work are included in Section 6.

2. Related Works

VM migration is the key operation for cluster management which received sustained attention in recent years [3–7]. Some research investigated the cost of migration. Mustafa and Michelle et al. studied the power cost of fog-enabled data center and proposed an efficient power model [4]. Jayamala and Valarmathi [8] introduced a decentralized platform to monitor the cost of migration while He and Buyya [7] proposed a thorough formulation for the cost of migration in their research. Other researches [9–11] focused on the optimization of migration scheduling. Wang et al. [10] introduced a new planning method for VM migration in software-defined networks. He et al. [11] studied the relationship between migration planning and SLA in SDN enabled clouds. Although these researches achieved substantial progress on specific point, one important factor they overlooked is how to judge whether a migration should be carried on with the related PM confronting high resource utilization. Different from the above studies, this paper is concerned about migration risk when the migration process in the scheduling queue is inevitable.

For live VM migration algorithm, some performance metrics should be fully concerned. Firstly, energy conservation is a common objective in optimization of VM migration. Osama et al. [5] investigated the trace simulation for migration and introduced a new energy-aware approach for live migration in cloud center. Beloglazov and Buyya [12] discussed the efficiency of energy consumption for VM consolidation and VM migration. Secondly, the performance of applications during migration is also a research hotspot. Research [7] concerned the performance evaluation of live migration. A series simulation is implemented to test the duration, downtime, and transferred data for migration. Mandal et al. [13] focused on SLA violation of migration and designed a new algorithm to decrease the performance degradation by leveraging optimal VM selection. Thirdly, there are researches focusing on other metrics. Research [14] used predictive techniques to pursue load balancing for live migration while study [15] innovated a new geometric distributed algorithm to optimize load status of multi-VM migration across different IDC. Focusing on the key performance metric of general application, both execution time and response time were involved in the corresponding experiment of this paper.

From the algorithm perspective, various algorithms were utilized to deal with different problems for live VM migration. Firstly, heuristic algorithms were leveraged in [18, 19] to solve simple objective. Research [18] investigated the 29-day trace of sampling data from Google cluster. Leveraging the curve line of workload fluctuation, a heuristic algorithm is proposed to adjust threshold for VM migration. Torre et al. [19] introduced a heuristic algorithm based on island population model to approximate the Pareto optimal of VM placement. Heuristic algorithm is usually efficient with low complexity. However, it is hard to deal with multiobjective problem which is common in complex cluster computing scenario. Secondly, a large number of bionics algorithms were applied to solve complex optimization issue. Research [20] used hybrid IEFWA/BBO algorithm to achieve the energy efficient program for virtual resource management while ant colony algorithm was involved in [21, 22] to optimize the process of live VM migration. However, the variation of both application type and workload intensity was not considered in [20–22]. Tremendous iteration is required in bionics algorithm and the derived parameter would be disabled if the statuses of computing cluster change. Thirdly, with the booming AI technology, machine learning algorithms have been the exciting solution of complex problem in this area. Embedding individualized machine learning algorithm was introduced in [23] to increase the accuracy of load prediction. Leveraging reinforcement learning algorithm, Peng et al. [24] designed a special approach to manage virtual resource in IaaS cloud. Machining learning algorithms in [23, 24] were both effective and efficient. However, utilization of AI skill is not a trivia. General or classic AI algorithm needs to be substantially modified according to the specific optimization problem. The accuracy increment in [23] depends on a specific monitoring system which is designed for collecting data in supervised learning. The classic DQN is modified and a specific algorithm DQN-TP is designed in [24] for vehicular service scenario. For live VM migration optimization, the vehicular network status in DQN-TP was analyzed online to choose the best destination of migration. Therefore, ordinary AI algorithms cannot be used directly to solve specific problems. For the VM migration decision issue, there is no general AI algorithm for the specific requirement on metric and objective in this paper.

As far as we know, for the general decision problem of live VM migration, this is the first study on risk evaluation with DQN. Leveraging DQN to support live VM migration is not a trivia while obvious technical gap exists from engineering perspective. For instance, there are a plenty of factors which involve in the risk evaluation of live migration. In addition, the optimization space (CPU utilization of computing nodes in cluster) is also too tremendous to traverse for iteration. Furthermore, the optimal data distribution is a 50–50 split between successful migration and failed experience. However, the valuable data for failure process of live VM migration is quite sparse in our experiments. Due to the technical gap above, a specific DQN algorithm was innovated while uncertain theory is also utilized to make risk integration for live VM migration. The contribution of this paper is the particular DQN design which is refined to support online decision for live VM migration. The combined utilization of DQN and uncertain theory in real industrial experiment is also one of the highlights in this paper.

3. Problem Formulation

Live VM migration is quite complex and the risk of migration process is related with many factors such as PM, VM, and application itself. In this paper, the migration risk is evaluated by the above 3-level framework and normalized index is formulated to quantify risk of live migration.

The overall time for live migration mainly includes migration duration and downtime which is at the end of migration process. Considering the application level, the foremost factor for performance is the corresponding downtime. Although different policies (precopy, postcopy, and lazy-copy) have different principles and effects, all the stop-and-copy approach will suspend the active service of application during migration downtime. Denote R_{ap} as the migration risk at application level; then it can be formulated as

$$R_{\mathrm{ap}} = \begin{cases} \alpha \dfrac{D_r}{N}, & M - \text{intensive}, \\ \\ 0, & \text{other application}. \end{cases} \tag{1}$$

As is shown in equation (1), α is the general coefficient while D_r is the dirty rate of memory coping during live migration which can be achieved by get_dirty_log. In the last copying round, the guest VM will suspend and the new dirty page will be transmitted over the network to the target VM. The length of downtime is proportional to D_r and inversely proportional to N which represents the assigned network bandwidth during transmission within downtime. In addition, the application type is also an important factor; e.g., some CPU-intensive jobs have little memory changes; thus the risk of downtime can be ignored.

In the second level, the migration time is the risk factor for VM involved in migration. Performance degradation of VM usually occurs if the migration time is too long [23]. Research [24] verified that the CPU utilization of domin-0 is inversely proportional with the migration time while Akoush et al. [25] leveraged the relationship between migration time and the process of precopy phase. Considering the above factors, the overall migration time t_i can be formulated as equation (2) while the risk at VM level is proportional with t_i using parameter β.

$$t_i = \begin{cases} \dfrac{t_{i-1} D}{f\left(U_{\mathrm{cpu}}, R\right)}, & i \geq 2, \\ \\ \dfrac{V_{\mathrm{mem}}}{f\left(U_{\mathrm{cpu}}, R\right)}, & i = 1. \end{cases} \tag{2}$$

As is shown in equation (2), V_{mem} represents the initial memory volume of VM before corresponding migration. The function of U_{cpu} and R (memory dirty rate) may differ with different VM type. In order to simplify this problem, the risk at VM level can be formulated as

$$R_{vm} = \begin{cases} \dfrac{\beta t_{i-1} D}{R U_{\mathrm{cpu}}}, & i \geq 2, \\ \\ \dfrac{\beta V_{\mathrm{mem}}}{R U_{\mathrm{cpu}}}, & i = 1. \end{cases} \tag{3}$$

Considering the computing node itself, severe performance degradation would appear along with computing resources shortage caused by contention between multiple jobs or VMs. Researches [26, 27] proved resource contention exists while performance inference grows with the ascending number of coming jobs or VM. Firstly, denote I_{sd} to be the performance interference from high utilization of computing resource. In this paper, I_{sd} has quadratic relationship with the utilization of multiple computing resources which is denoted as U_i in

$$I_{\mathrm{SD}} = \left(\sum_{i=1}^{p} \gamma_i U_i \right)^2. \tag{4}$$

The parameter γ_i in equation (4) reflects the importance or weight for each resource dimension. Comparing with collocation of different resource type, performance interference would be heavier if the resource demand comes from homogeneous jobs. Resource contention is taken into consideration while the corresponding interference factor I_{xy} is formulated in

$$I_{\mathrm{XY}} = \frac{\left\| p \sum X_i Y_i - \sum X_i \sum Y_i \right\|}{\sqrt{p \sum X_i^2 - \left(\sum X_i\right)^2} \sqrt{p \sum Y_i^2 - \sum Y_i^2}}. \tag{5}$$

In equation (5), p represents the number of resource dimensions while X and Y are different computing jobs. Note that I_{sd} and I_{xy} are mutually reinforcing factors; thus the risk at PM level is the product of I_{sd} and I_{xy}. The coefficient γ_i can be regarded as the same constant in order to simplify the formula. Then, the final risk at PM level R_{pm} can be represented as

$$R_{\mathrm{pm}} = \frac{\gamma U_i \left\| p \sum X_i Y_i - \sum X_i \sum Y_i \right\|}{\sqrt{p \sum X_i^2 - \left(\sum X_i\right)^2} \sqrt{p \sum Y_i^2 - \sum Y_i^2}}. \tag{6}$$

Migration time can be achieved by nova migration list to measure the duration of migration while the downtime of live migration could be measured by the timestamp difference in nova log files. Leveraging these monitoring tolls in cloud, the risk value at three levels (application, VM, and PM) can be achieved at real time. The remaining problem is how to deal with the quantitative relationship between these three values.

4. Algorithm Interpretation

With the growing scale and urgent requirement of precise control in IaaS computing cluster, AI technology is the dependent solution in the future. The intention of this paper is to make risk evaluation from AI perspective. As VM migration will have a close interaction with computing cluster, e.g., the source PM, the target VM, and the application itself, then the reinforcement learning algorithm is a suitable AI approach for its interaction between the individual and the environment.

4.1. Modified Deep Q Network. The basic Q learning is not suitable due to approximate infinite Q table composed of many related factors which have been formulated in Section 3. In this paper, deep Q network is applied to provide risk analysis at different level. Note that the classic deep Q learning network approach is modified and some innovative method is provided to deal with the specific risk problem in complex cluster environment.

In this paper, three sub-DQN networks are built to quantify this three-level risk system which is shown in Figure 1. Supervised learning for each monitoring point is excluded as the risk of migration from the computing cluster should be evaluated as a whole feedback. Note that the CPU utilization needs to be discretized to decrease the status optimization space. For instance, the continuous CPU utilization is converted to discrete values. In this paper, each three-percentage interval degenerates into one CPU point while the actual value will be identified as the value of the nearest discrete point. In DQN network, α, β, and γ can be replaced by the general parameters w and b while the loss will be minimized by coefficient adjustment in neural network. The related metric values for D_r, N, R, and U_{cpu} can be derived from monitoring tolls such as Prometheus or Grafana. These basic monitoring metrics are denoted as input values while the reward is derived from the deferred feedback at next monitoring point.

In order to sustain the stability of neural network, there are two DQN (target network and evaluation network) for each risk evaluation. In addition, medium memory bank is applied in this system. As migration with severe performance degradation is quite rare then memory bank is utilized to store the effective migration experience under high resource utilization. In this paper, the size of memory bank is set to be 100 while the batch sample size for DQN training is to be 10. Note that training begins only after 30 samples exist in memory bank. This approach avoids invalid training and blocks the correlation between the sampling data. The parameters in evaluation network are updated in each loop with batch sample data while the target network updating is taken in a deferred way. In this paper, the parameters of evaluation network will be assigned to the target network after each 50 learning steps. Then the target network can be adjusted in a moderate way. The computing loss will be transferred backward according to the difference between the actual feedback and the predictive value from evaluation network.

4.2. Risk Integration and Basic Steps. As the risk mechanism is quite complex and sampling data of failure migration is quite sparse, uncertainty theory is utilized to deal with this decision problem for VM migration in this paper. Uncertainty theory is an innovative approach to deal with indeterminacy besides probability theory. Innovated by Liu [28], uncertain theory is founded based on uncertain measure with normality axiom, duality axiom, subjectivity, and product axiom. Thorough formulation is provided in previous literature. For instance, Prof. Peng formulated the above four axioms of uncertain measure and provided clear optimization model for uncertain logistic network [29]. Different from the theory development in [29], it is utilization of uncertain theory from engineering perspective in which uncertain risk analysis is leveraged in risk integration.

Denote θ to be the uncertain measure function; then the overall risk can be formulated as equation (7). Liu declared that probability theory is applicable to model frequencies while uncertainty theory is applicable to model belief degrees [24].

$$R = \theta\left\{ \bigcup_{j=1}^{m} \left(R_j < \xi_j\right) \right\} = \bigvee_{j=1}^{m} \theta\left\{R_j < \xi_j\right\}. \tag{7}$$

In equation (7), risk is the uncertain variable while ε_j are constant or independent variables with regular uncertainty distributions. Compared with probability theory, one of the different principles for uncertain approach is to take the maximum rather than the product of the probabilities. Therefore, the final risk can be represented as equation (8) if the application is memory intensive. Otherwise, the equation can be converted to equation (9) for simplicity.

$$\begin{aligned} R &= \theta\left\{ \bigcup_{j=1}^{m} \left(R_j < \xi_j\right) \right\} \\ &= \theta\left(R_{ap} < \xi_1\right) \vee \theta\left(R_{vm} < c_2\right) \vee \theta\left(R_{pm} < c_3\right), \end{aligned} \tag{8}$$

$$\begin{aligned} R &= \theta\left\{ \bigcup_{j=1}^{m} \left(R_j < \xi_j\right) \right\} \\ &= \theta\left(R_{vm} < c_2\right) \vee \theta\left(R_{pm} < c_3\right). \end{aligned} \tag{9}$$

In equations (8) and (9), εj is uncertain variable which can be flexibly adjusted according to SLA while C2 and C3 are empirical constants. The ultimate risk R can be achieved by combining equations (1), (3), (6), (8), and (9). The

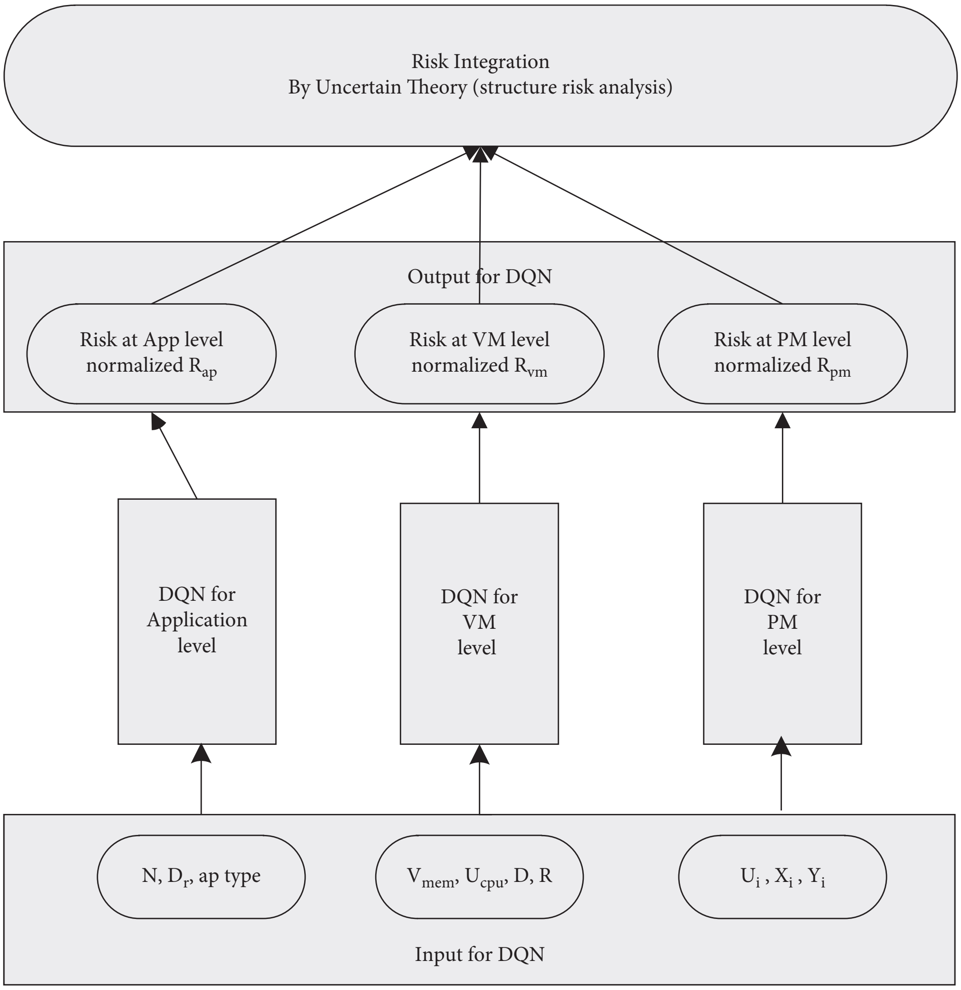
Risk Integration
By Uncertain Theory (structure risk analysis)
Output for DQN
Risk at App level normalized R_{ap}
Risk at VM level normalized R_{vm}
Risk at PM level normalized R_{pm}
DQN for Application level
DQN for VM level
DQN for PM level
N, D_r, ap type
V_{mem}, U_{cpu}, D, R
U_i , X_i , Y_i
Input for DQN

Figure 1: Structure of the risk evaluation system.

underlying principle is to obtain the maximum brief degree for risk among three levels.

Note that the uncertain theory [24] is different from the probabilistic approach. In addition, other counterpart approaches are provided in next section to evaluate the effect of different integration methods. Both the modified DQN and uncertain approach are involved in our algorithm and the core steps are shown as follows in Table 1.

5. Performance Evaluation

In this section, corresponding experiments are launched to evaluate the performance of algorithms.

5.1. Experimental Settings and Metrics. Related experiments can be carried out with different server type to implement heterogeneous cluster scenario; e.g., two basic types of server in our lab were applied: server A (Intel i79750H, hexa-core with 2600 MHz, 16 GB RAM) and server B (Intel octa-core i9-9900k with 3600 MHz, 32 GB RAM). Cloud management platforms, e.g., OpenStack, can be adapted to manage the virtual resource of cluster composed of the above two types of servers. Nova (15.1.1) is implemented here to undertake computing jobs in VM from control node and compute node in this cluster. In addition, the realization of our strategy depends on modifying related components (e.g., nova scheduler) of nova in OpenStack. To fit the real scenario in IaaS cloud, different types of VM are created based on different mirrors images.

The workloads applied in this paper are mixed types of benchmarks, e.g., SPECCPU, Netperf, and SPECweb2005. In addition, considering the importance of web application, ApacheBench test is also implemented to test the performance of response time which is also a crucial metric in performance evaluation. To increase the frequency of migration sampling data, extra workloads are implemented on PMs to bring about fluctuations of resource consumption. These workloads include database transactions, matrix transposition, and also a special probe designed to test the execution time for CPU-intensive application. Note that only the VM migration involved with resource utilization surpassing 60 percentages is regarded as qualified data which will be stored into memory bank of DQN. In this experiment the resource situation of the small cluster is monitored by Prometheus (2.8.1).

For experimental metric, two level metrics are considered in our experiments. Firstly, the precise and effective DQN is evaluated by the difference between target Q and evaluative

Table 1: DQN & uncertain integration for migration risk evaluation.

The core steps of DQN and uncertain integration for risk system
1. Initialize the DQN structure for risk at each level.
2. Configure the coefficient values such as learning rate and discount factor.
3. Choose the appropriate degree of discretion according to computing capacity.
4. Set the random selection method for action choice.
5. Create memory bank and start computing cluster.
6. Monitor metrics and store the transition.
7. While (step<=max_iter || accuracy is not satisfied).
8.   If step % 50 == 0.
9.   Assign the parameters to target network.
10.   Choose 10 samples from memory bank.
11.   Calculate the target value with actual feedback and discounted evaluative value.
12:     Obtain the loss and train the model with batch sampling data.
13.   Store the transition.
14.   step_counter+=1s.
15. Integrate three-level risk by searching maximum value from uncertain approach.
16. Evaluate migration decision by integrated ultimate risk.

Q. Note that the loss is not stable and the corresponding value may fluctuate dramatically due to different migrations. Therefore, CV (coefficient of variation) is introduced in this paper. The first metric is denoted as CV_j which is represented as equation (10). In addition, CV_j is related with the batch sampling data at corresponding level and the result value will be calculated and showed for each synchronization process between evaluation network and target network.

$$\mathrm{CV}_j = \frac{Q_{\mathrm{tar}} - Q_{\mathrm{eva}}}{Q_{\mathrm{tar}}}. \tag{10}$$

For SLAV metric, the response time is taken into consideration as it is the most obvious indicators for web applications. The response time data comes from the ApacheBench test in which client constantly sends requests to the Apache server for access to the homepage of the website in our experiment. In addition, the execution time is also considered as it is important metric for CPU-intensive jobs. A specific lightweight probe is utilized to derive the ratio between the ideal execution time and the actual value.

Other details of this research include parameter setting, discarded metrics, and unrevealed treatment for this experiment. The weighted value for three components in reward calculating is set to be equal in initial setting. Note that we just focus on the optimization of overloaded PM. Besides the metrics discussed above, downtime is also an important metric for migration. As the qualified migration sampling data is sparse, general downtime is not included in the metrics and we just keep the default value for related parameters such as max_downtime, steps, and delay for the performance evaluation in Section 5. B. In addition, the experiment is carried out in LAN network with high throughput; thus the network bandwidth has no direct affection on performance evaluation.

5.2. Comparative Algorithms and Experimental Results. In this subsection, we address the experimental results and give the relevant analysis. Three counterpart methods are compared with the uncertain approach in the performance evaluation. The difference of these methods is the way how to integrate risk index at different level. The first method is probabilistic approach in which the probability for overall risk is the product of risk index at three different levels. Assume that the R_{ap}, R_{vm}, and R_{pm} are obtained by DQN; the overall risk R can be derived according to equation (11) in which r means the reliability.

$$\begin{aligned} R &= 1 - r \\ &= 1 - r_{\mathrm{ap}} \cdot r_{\mathrm{vm}} \cdot r_{\mathrm{pm}} \\ &= 1 - \left(1 - R_{\mathrm{ap}}\right)\left(1 - R_{\mathrm{vm}}\right)\left(1 - R_{\mathrm{pm}}\right). \end{aligned} \tag{11}$$

In comparison, the ultimate risk R is the max of three risk values according to the theorems of uncertain approach as shown in equations (8) and (9). For counterpart methods' mean and median, the overall risk R is equal to the mean and median of the risk values at three levels.

Firstly, we evaluate the performance of DQN at different levels. The max iteration is set to be 500 and CV value is calculated every 50 iterations which is also the synchronization point between target DQN and evaluation DQN. As Figure 2 shows, the prediction inaccuracy decreased constantly for all risk values. The underlying reason is that the coefficients (weight and bias) of deep Q network are constantly adjusted during the training process. Leveraging the equations in Section 3, the initial accuracy is 64.4, 62.7, and 71.3 percentages for Ap, Vm, and Pm level while the final value is 92.9, 96.8, and 95.7 percentages, respectively. This showed reinforcement learning skills are reliable in risk evaluation. In addition, R_{vm} has the best DQN model at the end of training process while R_{ap} is not stable and has some deficiency in comparison. The reason is that the input for R_{ap} DQN is dependent on the application type which is random in this experiment.

The performance on execution time and response time is also analyzed to evaluate the uncertain approach and other risk integration methods. Figure 3 shows the relative execution time for sampling migration data. A lightweight probe is utilized to test the ratio between ideal execution time (application running on VM which exclusively uses the computing resource of PM) and actual execution time. Note that the number of qualified migration processes is different for different integration methods discussed above. The criterion in our experiment is that the migration will be launched only if the overall risk index is less than 50 percentages. All the qualified migration experiences in the memory bank are collected during the experimental duration for 100 consecutive minutes. The ratio of execution time is derived from the log files related with the qualified sampling data. As Figure 3 shows, only 12 experiences of VM migration were denoted while the qualified numbers in median and mean approach are 27 and 31, respectively. The uncertain method has moderate migration number and all the relative execution time is the mean value of

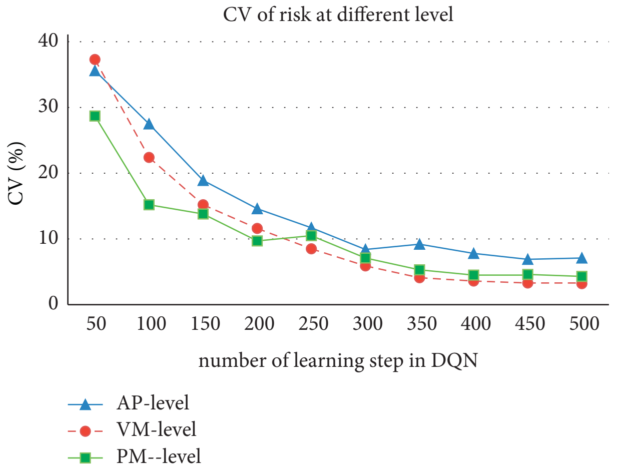

Figure 2: CV of risk at three levels.

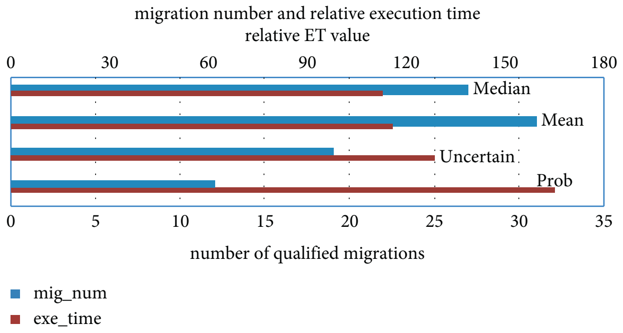

Figure 3: Migration number and execution time.

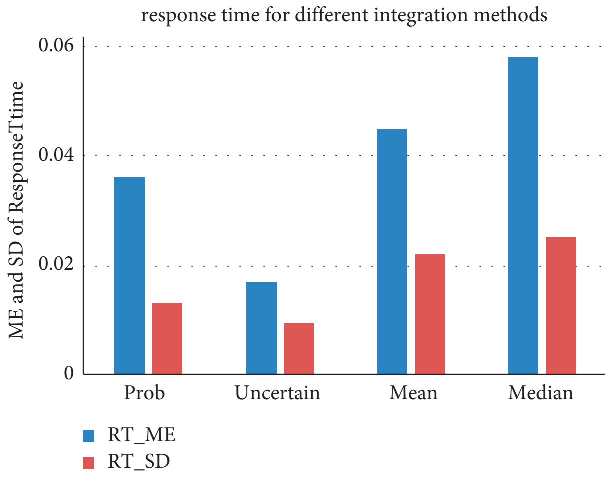

Figure 4: RT evaluation.

corresponding qualified migrations. Considering probability method, Figure 3 shows the ET metric delayed over 60 percentages while the delayed time for median and mean approach is less than 20 percentages. This reflects that probability method may overestimate the risk for small sampling event. On the other hand, the mean and median methods are aggressive algorithms for migration decision; thus the mean ET value is quite low. The underlying reason is that statistical probability is usually inconsistent with the frequency in real application scenario. In this paper, the overall risk may be overestimated as the monitoring data for failure migration experience is quite sparse. In the contrary, the uncertain approach provides unique method to deal with the uncertain reliability function or risk analysis. The quality of dataset is crucial in training process for machine learning. In the experiment of this paper, the optimal data distribution is a 50–50 split between successful VM migration and failed VM migration. However, it can not be achieved due to the limitation of experiment—some of failed migration will induce server crash and all the running applications in each VM will halt with abnormal termination. Therefore, the percentage of failed migration is quite low in the experiment of this paper. Uncertain theory is suitable for uncertain events (failed migration with heavy performance degradation) with small sampling data.

Response time is another crucial metric for web applications. In our experiment, response time is monitoring by AB test commends which are inherent in Apache server. Corresponding mean value and standard deviation are calculated based on the corresponding migration experience within the experimental duration. As Figure 4 shows, for RS metric, uncertain method dominates other approaches on both ME and SD value. For mean and median, aggressive methods have obvious higher deviation. Figure 4 shows RS time is higher if the migration is launched frequently (mean and median approach). In addition, SD value is also increased due to overmigration in mean and median approach which is shown in Figure 4. One of the interesting discoveries is that RS time is also deferred if target VM cannot be migrated timely. According to the analysis above, the overall risk is overestimated for probability approach. As the migration time is delayed, the resource contention of computing nodes may become more serious. The resource contention would extend the downtime of migration which directly affects the RT performance of probability approach. Monitoring data in this experiment verified some extreme RS values (more than 0.1 seconds) if the target VM is confronting high resource utilization. In comparison, the uncertain approach has obvious advantages due to moderate migration policy. Corresponding experimental analysis shows that our uncertain approach is a moderate way to deal with the risk integration at different levels. Aggressive and unapt migration decision may lead to severe performance degradation.

6. Conclusion and Future Works

VM migration is widely used for its convenience, yet the risk of live migration has not received enough attention. Performance degradation may occur due to improper risk estimation. Focusing on the risk issue for live migration, this paper provides innovative approach from AI insight. Leveraging reinforcement learning, DQN is modified in this paper to train the prediction accuracy while uncertainty theory is leveraged to provide specific integration for risk system. Series experiments showed the DQN is reliable while uncertain approach is a moderate integration way for training with sparse sampling data.

One direction of our future work is to introduce the topology of networks in a hybrid computing cluster. In complex cluster computing especially for hybrid network environment, the status of network fluctuates dynamically and network congestion sometimes occurs with uncertain. Considering the complex dynamic [25–29] graph optimization (the weight of each path is variable rather than constant), the route selection would greatly increase the complexity of live VM migration. These challenges are beyond current work in this paper.

Data Availability

The data used to support the findings of this study have been deposited in the Specweb repository ([http://www.spec.org/web2005/]).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Key Research and Development Program (2018YFB1306000), in part by the National Natural Science Foundation of China under Grants 61802455 and 62172457, in part by postdoctoral funding projects under Grant ZJ2020074, in part by Key Research and Development Program of Science and Technology in Henan with title "Research on Cache Consistency of Heterogeneous Multicore High Performance Processors," and in part by key research and development program of science and technology in Henan under Grants 202102210139 and 202102210098.

References

[1] T. Duong-Ba, T. Tran, T. Nguyen, and B. Bose, "A dynamic virtual machine placement and migration scheme for data centers," *IEEE Transactions on Services Computing*, vol. 14, no. 2, pp. 329–341, 2021.

[2] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 11–31, 2013.

[3] M. Aldossary, "A review of dynamic resource management in cloud computing environments," *Computer Systems Science and Engineering*, vol. 36, no. 3, pp. 461–476, 2021.

[4] I. K. Mustafa and M. Z. Michelle, "Adaptive virtual machine migration based on performance-to-power ratio in fog-enabled cloud data centers," *The Journal of Supercomputing*, vol. 77, 2021.

[5] A. Osama, F. Matthew, and T. Nigel, "Using virtual machine live migration in trace-driven energy-aware simulation of high-throughput computing systems," *Sustainable Computing-Informatics & Systems*, vol. 29, 2021.

[6] P. Xiao, Z. Ni, D. Liu, and Z. Hu, "A power and thermal-aware virtual machine management framework based on machine learning," *Cluster Computing*, vol. 24, no. 3, pp. 2231–2248, 2021.

[7] T. He and R. Buyya, "Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 55–68, 2019.

[8] R. Jayamala and A. Valarmathi, "An enhanced decentralized virtual machine migration approach for energy-aware cloud data centers," *Intelligent Automation & Soft Computing*, vol. 27, no. 2, pp. 347–358, 2021.

[9] C. Hu, Y. Deng, G. Min, P. Huang, and X. Qin, "QoS promotion in energy-efficient datacenters through peak load scheduling," *IEEE Transactions on Cloud Computing*, vol. 9, no. 2, pp. 777–792, 2021.

[10] H. Wang, Y. Li, Y. Zhang, and D. Jin, "Virtual machine migration planning in software-defined networks," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 1168–1182, 2019.

[11] T. He, A. N. Toosi, and R. Buyya, "SLA-aware multiple migration planning and scheduling in SDN-NFV-enabled clouds," *Journal of Systems and Software*, vol. 176, Article ID 110943, 2021.

[12] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1366–1379, July 2013.

[13] R. Mandal, M. K. Mondal, and S. Banerjee, "An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing," *The Journal of Supercomputing*, vol. 76, no. 1, 2020.

[14] L. H. Hung, C. H. Wu, and C. H. Tsai, "Migration-based load balance of virtual machine servers in cloud computing by load prediction using genetic-based methods," *IEEE Access*, vol. 9, no. 99, p. 1, 2021.

[15] G. Singh and A. K. Singh, "Optimizing multi-VM migration by allocating transfer and compression rate using geometric programming," *Simulation Modelling Practice and Theory*, vol. 106, Article ID 102201, 2021.

[16] R. Norfazlin and Y. U. Kalsom, "Literature survey: statistical characteristics of google cluster trace," in *Proceedings of the 2018 4th International Conference on Advances in Computing, Communication and Automation (ICACCA 2018)*, Subang Jaya, Malaysia, October 2018.

[17] E. Torre, J. J. Durillo, V. de Maio et al., "A dynamic evolutionary multi-objective virtual machine placement heuristic for cloud data centers," *Information and Software Technology*, vol. 128, 2020.

[18] H. M. Ali and C. Lee Daniel, "Optimizing the energy efficient VM placement by IEFWA and hybrid IEFWA," *BBO Algorithms" Simulation Series*, vol. 48, no. 8, pp. 61–68, 2016.

[19] S. G. Sutar, P. J. Mali, and A. Y. MoreAmruta, "Resource utilization enhancemnet through live virtual machine migration in cloud using ant colony optimization algorithm," *International Journal of Speech Technology*, vol. 23, no. 1, pp. 79–85, 2020.

[20] M. Mahil and T. Jayasree, "Combined particle swarm optimization and ant colony system for energy efficient cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 10, 2021.

[21] S. Mashhadi Moghaddam, M. O'Sullivan, C. Walker, S. Fotuhi Piraghaj, S. Fotuhi Piraghaj, and C. P. Unsworth, "Embedding individualized machine learning prediction models for energy efficient VM consolidation within cloud

data centers," *Future Generation Computer Systems*, vol. 106, pp. 221–233, 2020.

[22] J. Peng, C. Wang, J. Fu, G. U. Xin, M. U. Yueyue, and W. Liu, "A fast deep Q-learning network edge cloud migration strategy for vehicular service," *Journal of Electronics and Information Technology*, vol. 42, no. 1, pp. 58–64, 2020.

[23] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "CloudScale: elastic resource scaling for multi-tenant cloud systems," in *Proceedings of the ACM Symposium on Cloud Computing*, pp. 51–64, Cascais, Portugal, October 2011.

[24] Y. Wu and M. Zhao, "Performance modeling of virtual machine live migration," in *Proceedings of the IEEE International Conference on Cloud Computing*, pp. 492–499, Washington, DC, USA, July 2011.

[25] S. Akoush, R. Sohan, A. Rice, and A. W. Moore, "Predicting the performance of virtual machine migration," in *Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 37–46, Miami Beach, FL, USA, August 2010.

[26] R. Iyer, R. Illikkal, O. Tickoo, L. Zhao, P. Apparao, and D. Newell, "VM 3: measuring, modeling and managing VM shared resources," *Computer Networks the International Journal of Computer & Telecommunications Networking*, vol. 53, no. 17, pp. 2873–2887, 2009.

[27] J. Mukherjee, D. Krishnamurthy, and J. Rolia, "Resource contention detection in virtualized environments," *IEEE Transactions on Network & Service Management*, vol. 12, no. 2, pp. 217–231, 2015.

[28] B. Liu, *Uncertainty Theory*, Springer, Berlin, Germany, 4Eds edition, 2015.

[29] J. Peng, "Mathematical models for logistics network optimization with uncertain data," in *Proceedings of the 2019 International Conference on Information Technology and Computer Communications*, pp. 93–100, Singapore, August 2019.