2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)

Bayesian networks-based selection algorithm for virtual machine to be migrated

Chengyu Yan
Department of Computer science and technology
Jiangnan University
Wuxi, China
cheneyyin@hotmail.com

Zhihua Li
Department of Computer science and technology
Jiangnan University
Wuxi, China
zhli@jiangnan.edu.cn

Xinrong Yu
Department of Computer science and technology
Jiangnan University
Wuxi, China
espresso_yu@163.com

Ning Yu
Department of Computer Science
Georgia State University
Georgia, USA
nyu4@student.gsu.edu

***Abstract*—In cloud data centers, virtual machine (VM) consolidation is one of the challenge topics. In which, the selection of VMs to be migrated is one of the key issues in the process of VM consolidation. In this paper, under consideration of the dynamical uncertain environment, a Bayesian networks-based estimation model was constructed. Because excessive VM migrations influence the Quality of Service (QoS) of data center, the model aims at estimating the migration probability of VMs and calculating the potential total number of migrations occurred in physical hosts. Based on the proposed model, a Bayesian networks-based selection algorithm (BN-SA) for VMs to be migrated was proposed. The BN-SA adaptively adjusts the overloaded threshold and selects VMs which have relatively short migration time and big impact on potential migrations of host in the phase of reallocating VMs. The experimental results show that BN-SA algorithm has a promising performance.**

Keywords—Cloud Computing; VM selection; VM migration; Bayesian Networks

I. INTRODUCTION

High energy consumption always is a huge challenge to resource management of data center, along with the scale enlargement of data centers, and this problem becomes more and more serious [1]. Research report of IBM pointed out that the average CPU utilization of physical hosts is just 15%~20% [2]. And [3] stated that the physical host in idle state consumes 70% energy of its peak consumption. Obviously, many physical hosts are in idle that will cause low energy efficiency and huge waste of the resource. Currently, VM migration technology was used in VM consolidation in practice. The resource manage system migrates VMs from underutilized hosts employing migration technology, and then shuts down or switches these hosts to the sleep mode that will optimize resource usage and reduce energy consumption.

As one of the energy saving technologies, virtualization technology has been used and researched widely. Cloud service providers can create multiple VMs in a single physical host and adjust allocation relationship between VMs and physical hosts through virtualization technology. According to current resource requirements of VMs, VMs can be consolidated to the minimal number of physical hosts by using live migration [4-6]. So the consolidation can decrease the scale of active physical hosts. However, due to the variable workload of data centers, the requirements of VMs may rise, then the excessive consolidation may result in violating the service level agreement (SLA) and lead to poor QoS. In conclusion, the main challenge of dynamic VM consolidation is keeping balance between ensuring QoS and saving energy.

Usually, dynamic VM consolidation includes four phases in [7]: (1) determining when a host is considered as being overloaded requiring migration of one or more VMs from this host; (2) determining when a host is considered as being underloaded leading to a decision to migrate all VMs from this host and switch the host to the sleep mode; (3) selection of VMs that should be migrated from an overloaded host; and (4) finding a new placement for the selected VMs that will to be migrated. It needs to be emphasized that migrated VMs need to stop service and the long duration of migration may influence the QoS of data centers. So reducing unnecessary VM migrations can enhance the QoS. On the other hands, [8] pointed out that lots of live migrations for VMs were not recommended, because which will cost more extra computing resource and increase uncertain workload as well as result in the poor QoS and SLA violation in data centers.

Several research works have been done on this topic. In [9-11], they optimize an overloaded threshold from different perspective respectively, and then some VMs are migrated from hosts which will be overloaded according to the present threshold. But there exist a common problem that it is easy to result in excessive migration. A lot of unnecessary VM migrations may cause the poor QoS in data centers. Three adaptive overloaded host detection algorithms have been

978-1-5090-3936-4/16 $31.00 © 2016 IEEE
DOI 10.1109/BDCloud-SocialCom-SustainCom.2016.89

proposed in [7]: Median Absolute Deviation (MAD), Interquartile Range (IQR) and Local Regression (LR). And three VM migration algorithms have been proposed in [7]: Minimum Migration Time (MMT), Maximum Correlation (MC) and Random Selection (RS). The MMT algorithm migrates VMs with minimum migration time preferentially, the MC algorithm migrates VMs with highest correlation of CPU utilization with other VMs allocated on the host preferentially and the RS algorithm migrates VMs randomly. And [7] also proposed the Power Aware Best Fit Decreasing (PABFD) algorithm to reallocate VMs to be migrated. The PABFD sorts all the VMs on a decreasing order by their current CPU utilizations and allocates each VM to a host which can provide the least increment of the power consumption after the VM placed on it. However, it is easy to result in that some hosts with high load and suffer from poor QoS, and some hosts with low load and suffer from energy waste. Namely, it is hard to keep a promising balance between the QoS and energy utilization.

In this paper, considering the variability of workload, we discuss a detail issue in the VM consolidation field. How to select VMs to be migrated from physical hosts? For this issue, we build a Bayesian Networks-based estimation model, which can estimate the migration probability of VMs and the potential total times of migration occurred in physical hosts. Then we present Bayesian networks-based selection algorithm (BN-SA) for VMs to be migrated. The BN-SA estimates the potential probability of VM migration and the possible total times of VM migration. According to the overloaded probability of physical hosts, BN-SA adaptively adjusts the overloaded threshold and selects the VMs which have relatively short migration time and big impact on the potential total migrations of VMs allocated on overloaded hosts. The experimental results show that BN-SA can efficiently improve QoS and reduce VM migrations while decreasing energy consumption.

II. THE BAYESIAN NETWORKS-BASED ESTIMATION MODEL

We consider a data center consisting of n physical hosts. $H=\{h_1,h_2,\cdots,h_n\}$ is the set of physical hosts, and $VM_i=\{vm_1,vm_2,\cdots,vm_m\}$ is the set of VMs deployed on h_i. The CPU capacity of vm_j is denoted as c_j, r_j is the CPU capacity requested by vm_j, and d_j represents CPU demand of vm_j as a percentage. Equation (1) expresses the relation of c_j, d_j and r_j. The demand of h_i for CPU resource is denoted as a random variable D_i, aggregated by VMs allocated on h_i, and can be calculated by (2). a_j is defined as the CPU capacity that the host allocates to vm_j, C_i is the CPU capacity of h_i, u_i represents CPU utilization (in percentage) of h_i and defines as (3).

$$r_j = d_j \cdot c_j \tag{1}$$

$$D_i = \sum_{vm_j \in VM_i} r_j \tag{2}$$

$$u_i = \frac{1}{C_i} \sum_{vm_j \in VM_i} a_j \tag{3}$$

Due to dynamic workloads of VMs, it is hard to model the VM consolidation process. Bayesian networks (BN), a modeling method in complex uncertainty system, can reduce the difficulty of knowledge accessing and complexity of probabilistic reasoning effectively [12]. This paper models a BN to estimate the probability of VM migrations as shown in Fig.1. The model can estimate migration probability of the VM according to load patterns of the host which the VM deployed on and probability relationship between nodes in BN. The BN proposed in this paper contains 9 nodes and each of them is expressed as follow.

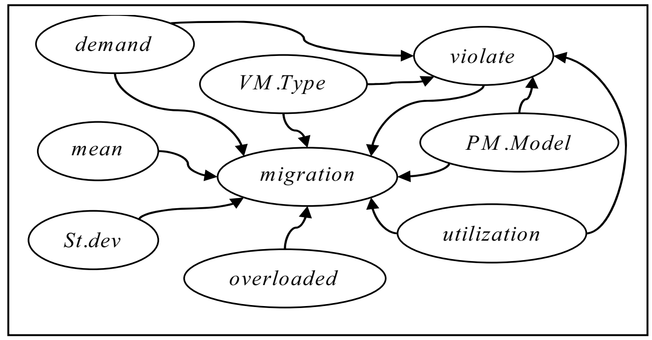

Fig. 1. The bayesian networks-based estimation model.

Node '*VM.Type*' (T) is the type of VM instance, different types of instance with different specifications of CPU and memory. Node '*PM.Model*' (M) is the model of physical hosts, for example, HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores × 1860 MHz, 4 GB). Node '*demand*' (d) represents the current CPU demand expressed in percent, node '*mean*' (m) and '*St.dev*' (sd) represent mean and standard deviation of recent resource demands of VMs respectively. According to that the resource demand of VMs follows normal distribution proposed in [13] and [14], three nodes ('*demand*', '*mean*' and '*St.dev*') are considered to describe workload patterns of VMs. Node '*utilization*' (u) represents the CPU utilization (in percentage) of physical hosts. Node '*violate*' (v) represents the violation of VM requirement. For example, if $a_j < r_j$ then $v = true$, it means that requirement of vm_j has not been satisfied. Generally, violations are relative to current VM requirements and the workload of physical hosts, so '*demand*', '*utilization*', '*VM.Type*' and '*PM.Model*' should be seen as parent nodes of '*violate*'.

Node '*overloaded*' (o) is the overloaded probability of physical hosts. We assume that the CPU demand of vm_j follows $d_j \sim N(\mu_j,\sigma_j^2)$. D_i is the workload of h_i and aggregated by workloads of all VMs allocated on it, so D_i follows $D_i \sim N(\mu_i,\sigma_i^2)$. Where μ_i and σ_i can be calculated by (4) and (5) respectively. When $D_i > C_i$, the demand of h_i exceeds its CPU performance, and this host will be overloaded inevitably. So the overloaded probability of h_i is defined as (6), where Φ is the probability distribution function of normal distribution. Node '*utilization*' and '*overloaded*' are used to describe the load state of physical hosts.

$$\mu_i = \sum_{vm_j \in VM_i} c_j \cdot \mu_j \tag{4}$$

$$\sigma_i = \sqrt{\sum_{vm_j \in VM_i} \left(c_j \cdot \sigma_j\right)^2} \tag{5}$$

$$P_{\text{over}}^i = \Pr\left(D_i > C_i\right) = 1 - \Pr(D_i \le C_i) = 1 - \Phi\left(\frac{C_i - \mu_i}{\sigma_i}\right) \tag{6}$$

Node '*migration*' (*mig*) represents the migration of VMs, if a VM is migrated then $mig = true$, otherwise $mig = false$. To estimate the migration probability of VMs with different demand patterns under physical hosts with different load states, we set these above 8 nodes as parent nodes of '*migration*'.

III. THE BN-BASED SELECTION ALGORIGTHM

A. The Probability Estimation Of VM Migration

d, m, sd, u and o introduced in section II are continuous stochastic variables, but they need to be discretized for estimating probability in BN. Value ranges of these stochastic variables can be divided into m bins, these bins are defined in (7), in which all of value ranges of those variables are $[0,1]$.

$$B_1 = \left[0, \frac{1}{m}\right), \quad B_2 = \left[\frac{1}{m}, \frac{2}{m}\right), \quad \cdots \quad, B_m = \left[\frac{m-1}{m}, 1\right] \tag{7}$$

$$f_B\left(x\right) = \sum_{b=1}^{m} b \cdot I\left(x \in B_b\right) \tag{8}$$

Equation (8) defines the function f_B which can map values to the relative bin, where $I(x \in B_b)$ is used to judge whether x is belong to B_b, if $x \in B_b$ then $I(x \in B_b) = 1$, otherwise $I(x \in B_b) = 0$. According to maximum likelihood estimation, the probability distribution of nodes in bayesian networks can be determined by observations. We assume X_i is one node of the network and R_i is the value set of it, then its parent nodes has q_i states. The conditional probability of $X_i = k$ is defined as (9), when the state of parent nodes is j,

$$\Pr\left(X_i = k \mid \pi\left(X_i\right) = j\right) = \begin{cases} \dfrac{e_{ijk}}{\sum_{k \in R_i} e_{ijk}}, & \sum_{k \in R_i} e_{ijk} > 0 \\ \dfrac{1}{\left|R_i\right|}, & otherwise \end{cases} \tag{9}$$

where e_{ijk} is the number of samples which meet $X_i = k$ and $\pi(X_i) = j$ in observations, and $\sum_{k \in R_i} e_{ijk}$ is the number of samples which meet $\pi(X_i) = j$. The migration probability of vm_j allocated on h_i can be estimated by (10). As show in (11), function $f_{\pi(v)}$ maps the observations of parent nodes $\pi(v) = (d, u, T, M)$ to corresponding states. The function $f_{\pi(mig)}$ do the same work like $f_{\pi(v)}$.

$$\begin{aligned} &P_{\text{mig}}\left(d_j, \mu_j, \sigma_j, T_j, u_i, P_{\text{over}}^i, M_i\right) \\ &= \Pr\left(v = true \mid f_{\pi(v)}\left(d_j, T_j, u_i, M_i\right)\right) \\ &\cdot \Pr\left(mig = true \mid f_{\pi(mig)}\left(v = true, d_j, \mu_j, \sigma_j, T_j, u_i, P_{\text{over}}^i, M_i\right)\right) \\ &+ \Pr\left(v = false \mid f_{\pi(v)}\left(d_j, T_j, u_i, M_i\right)\right) \\ &\cdot \Pr\left(mig = true \mid f_{\pi(mig)}\left(v = false, d_j, \mu_j, \sigma_j, T_j, u_i, P_{\text{over}}^i, M_i\right)\right) \end{aligned} \tag{10}$$

$$\begin{aligned} &f_{\pi(v)}\left(d_j, T_j, u_i, M_i\right) \\ &= \left(d = f_B(d_j), T = T_j, u = f_B(u_i), M = M_i\right) \end{aligned} \tag{11}$$

B. Adaptive Host Overloaded Detection

Algorithm 1:EOP	
1:	**Input:** h_i
2:	**Output:** $isOverloaded_i$
3:	Use (6) calculate P_{over}^i
4:	Use (12) calculate T_{u}^i
5:	**if** $u_i > T_{\text{u}}^i$ **then**
6:	$isOverloaded_i \leftarrow true$
7:	**else**
8:	$isOverloaded_i \leftarrow false$
9:	**end if**

As a result of variable workloads of cloud data centers, the host overloaded detection algorithm based on static threshold cannot adjust reserved resource according to variable workloads dynamically, and might allocate resources of data centers unreasonably. In order to cope with this problem, Estimation of Overloaded Probability (EOP) algorithm was proposed and used to detect physical hosts with overloaded risk. The current resource utilization of host can reflect the recent load level, and the overloaded probability measures the potential overloaded probability of host with current load pattern. In order to consider the two factors comprehensively, we proposed two assumptions: (1) when the overloaded probability is 0, the threshold will be set 100% and none VMs need to be migrated allocated on the host; (2) when the overloaded probability is 100%, the host will be not overloaded if the utilization is less than one specific value. In conclusion, the threshold T_{u}^i of host h_i is defined as (12),

$$T_{\text{u}}^i = 1 - s \times P_{\text{over}}^i \tag{12}$$

where parameter s represents the confidence of overloaded probability and can weighs the relationship between resource utilization and QoS. When s is relatively big, T_{u}^i will be set down drastically for the rise of P_{over}^{i} that make sure QoS. On the other hand, s is relatively small, T_{u}^{i} will keep high-level that tend to improve energy efficiency. If the CPU utilization of h_i exceeds the threshold, h_i should be seen overloaded and some VMs have to be migrated from it. EOP algorithm is described in Algorithm 1.

C. Selecting the Virtual Machine to be Migrated

When a physical host is considered to be overloaded, in order to improve QoS of the host, some VMs need to be migrated to reduce overloaded probability. Migrated VMs need to stop service that impact QoS. In order to reduce migration time, VMs which have shorter migration time relatively to the other VMs allocated to the host should be migrated preferentially. After a VM has been migrated, the CPU utilization, the overloaded probability of the host and the migration probability of remaining VMs allocated to the host will change. According to the migration probability estimation method by bayesian networks defined in section III part A, the total potential migration times of h_i can be calculated by (13), after the host h_i migrated the vm_k. In (13), u_i^{-k} represents the CPU utilization of h_i, denoted as (14). P_{over}^{i-k} is the overloaded probability after h_i migrated vm_k, denoted as (15). After h_i migrated vm_k, the average of recent requirements of resource and its standard deviation are calculated by (16) and (17) respectively.

$$M_i^{-k} = \sum_{vm_j \in VM_i - \{vm_k\}} P_{\text{mig}}\left(d_j, \mu_j, \sigma_j, u_i^{-k}, P_{\text{over}}^{i-k}\right) \cdot 1 \qquad (13)$$

$$u_i^{-k} = \frac{1}{C_i} \sum_{vm_j \in VM_i - \{vm_k\}} a_j \qquad (14)$$

$$P_{\text{over}}^{i-k} = 1 - \Phi\left(\frac{C_i - \mu_i^{-k}}{\sigma_i^{-k}}\right) \qquad (15)$$

$$\mu_i^{-k} = \sum_{vm_j \in VM_i - \{vm_k\}} c_j \cdot \mu_j \qquad (16)$$

$$\sigma_i^{-k} = \sqrt{\sum_{vm_j \in VM_i - \{vm_k\}} \left(c_j \cdot \sigma_j\right)^2} \qquad (17)$$

To reduce the following VM migrations, VMs with lower M_i^{-k} relatively to the other VMs allocated to the host should be migrated preferentially. Synthesizing migration time and the potential VM migrations on the host, the criterion of VM selection is defined in (18).

$$g_{\text{M}}^{i-k} = \frac{ram_k}{net_i} + \alpha \cdot M_i^{-k} \qquad (18)$$

In (18), ram_k is the amount of memory utilized by vm_k, and net_i is the bandwidth spared by h_i. α is the weight of M_i^{-k} relatively to migration time. Based on the above content, the VM selection algorithm based on BN is proposed. The BN-SA uses EOP to detect the host firstly, then BN-SA will migrate VMs with minimize criterion of VM selection from overloaded hosts preferentially. The BN-SA is described in algorithm 2.

Algorithm 2: BN-SA

```
1:  Input: h_i
2:  Output: VM_m
3:  VM_m ← ∅
4:  while EOP(h_i) = true do
5:     vm_mig ← ∅
6:     g_min ← MAX
7:     for all vm_k ∈ VM_i do
8:        Use (18) calculate g_M^{i-k}
9:        if g_M^{i-k} < g_min then
10:          g_min ← g_M^{i-k}
11:          vm_mig ← vm_k
12:       end if
13:    end for
14:    VM_m ← VM_m ∪ {vm_mig}
15:    VM_i ← VM_i − {vm_mig}
16: end while
```

IV. SIMULATION AND ANALYSIS

A. Simulation Environment

TABLE I. WORKLOAD DATA CHARACTERISTICS (CPU UTILIZATION).

Date	*Number of VMs*	*Mean (%)*	*St. dev. (%)*
03/03/2011	1052	12.31	17.09
06/03/2011	898	11.44	16.83
09/03/2011	1061	10.70	15.57
22/03/2011	1516	9.26	12.78
25/03/2011	1078	10.56	14.14
03/04/2011	1463	12.39	16.55
09/04/2011	1358	11.12	15.09
11/04/2011	1233	11.56	15.07
12/04/2011	1054	11.54	15.15
20/04/2011	1033	10.43	15.21

The experiment simulated a data center consisting of 800 heterogeneous physical hosts in CloudSim [15], these physical hosts are divided into two categories: Hp ProLiant ML110 G4 (Intel Xeon 3040 2cores 1860MHz, 4GB) and Hp ProLiant ML110 G5 (Intel Xeon 3075 2cores 2260MHz, 4GB). In this experiment, there are 4 types of VMs: High-CPU Medium Instance (2500MIPS, 0.85GB), Extra Large Instance (2000MIPS, 3.75 GB), Small Instance (1000MIPS, 1.7GB) and Micro Instance (500 MIPS, 613 MB). Workloads used in the experiment are 10 days real world trace from PlantLab [16], which were used by previous experiments [7]. These are stored in simple text files: one file per VM, in which each line contains a CPU load, as a percentage of the requested capacity. The characteristics of the workload for each day are shown in Table I.

B. Performance evaluation

In order to evaluate the algorithm performance reasonably, we used four metrics proposed in [7]: SLA Violations (SLAV), Energy Consumption (EC), VM Migrations (VMM) and ESV (Energy and SLA Violations). SLAV is used to evaluate QoS of data center which is defined in (19),

$$SLAV = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{T_i^{\mathrm{s}}}{T_i^{\mathrm{a}}}\right)\times\left(\frac{1}{m}\sum_{j=1}^{m}\frac{C_j^{\mathrm{d}}}{C_j^{\mathrm{r}}}\right) \tag{19}$$

where T_i^{s} is the total time during which the h_i has experienced the utilization of 100% leading to a SLAV [7]. T_i^{a} is the running time of h_i. C_j^{d} is unsatisfied CPU required capacity of vm_j caused by migration. C_j^{r} is CPU capacity required by vm_j. n and m represent the number of physical hosts and VMs respectively. ESV is used to evaluate overall performance both of energy consumption and QoS which is defined in (20).

$$ESV = EC \times SLAV \tag{20}$$

VMs which are migrated will stop service and influence QoS. So reducing unnecessary migrations can enhance QoS. Otherwise, frequent migration may lead to network congestion.

C. Results analysing

CloudSim [15] implemented some VM dynamic consolidation approaches, including four overloaded host detection algorithms (Single Threshold (ST) in [9], MAD, IQR and LR) and three VM selection policies (MMT, MC and RS). According to research in [7] and [9], we set parameter for overloaded detection algorithms as follows: MAD-2.5, IQR-1.5, LR-1.2 and ST-0.8. The parameter s of BN-SA is 1.0, α is 0.4. This experiment compared the BN-SA and other 12 combinations of detection and selection algorithms. All methods tested in the experiment use PABFD as VM placement algorithm.

The summary of simulation results is shown in TABLE II, in which value behind labels (MAD-MMT, LR-MMT and IQR-MMT etc.) represents their parameter respectively. According to TABLE II, concerning energy consumption, combinations which use LR as detection algorithm reduce more energy consumption than the other combinations. ST-MMT-0.8 consumes most energy in all tested algorithms. BN-SA delivers the best result in saving energy, and LR-MC-1.2 and LR-RS-1.2 are second to BN-SA. Compared to LR-MC-1.2 and LR-RS-1.2, BN-SA is only 82.1 percent of them in energy consumption. According to SLAV metric in TABLE II, BN-SA delivers the best result, MAD-MMT-2.5, IQR-MMT-1.5 and ST-MMT-0.8 are second to BN-SA, and the SLAV of BN-SA is 79.9 percent of the three combinations. The SLAV of LR-RS-1.2 is the highest in all tested algorithms. The SLAV of BN-SA is only 33.9 percent of LR-RS-1.2. So the BN-SA plays a good effect at enhancing QoS. Concerning VMM, BN-SA triggered less VM migrations than other algorithms. Though MAD-MC-2.5 and IQR-MC-1.5 are closed to it, the migration times of BN-SA are only 90.3 percent of them. It is obviously to know that BN-SA can decrease VM migrations effectively and avoid network congestion caused by frequent migration. Considering to the last evaluation metric ESV, BN-SA shows the best performance in all test algorithms, its ESV only 52.6 percent of MAD-MMT-2.5. MC and RS lose balance between energy consumption and QoS guarantee, and have worse performance than BN-SA. In conclusion, BN-SA proposed in this paper show the best in energy consumption, SLAV, VMM and ESV in all tested algorithms. BN-SA can keep a balance between energy consumption and ensuing QoS and enhance energy efficiency while reducing unnecessary migration.

TABLE II. THE SUMMARY OF SIMULATION RESULTS

Method	*EC(kWh)*	*SLAV (%)*	*VMM*	*ESV (%)*
BN-SA	**121.97**	**0.002648**	**21142**	**0.3230**
MAD-MMT-2.5	183.50	0.003348	26305	0.6143
MAD-MC-2.5	173.79	0.007111	23420	1.2358
MAD-RS-2.5	174.86	0.007038	23642	1.2307
LR-MMT-1.2	161.87	0.004974	28175	0.8051
LR-MC-1.2	148.51	0.007609	23931	1.1230
LR-RS-1.2	147.67	0.007807	23659	1.1528
IQR-MMT-1.5	187.53	0.003288	26497	0.6166
IQR-MC-1.5	177.70	0.006805	23394	1.2093
IQR-RS-1.5	178.61	0.006865	23740	1.2262
ST-MMT-0.8	188.50	0.003371	26602	0.6354
ST-MC-0.8	179.38	0.006989	23962	1.2537
ST-RS-0.8	180.43	0.006970	24155	1.2576

In order to analyze BN-SA in detail, we compare BN-SA to other 12 combinational algorithms. Fig.2 shows the variability of the number of active hosts along with time of consolidation. These algorithms all shut down many hosts in the former 25 periods and keep the number of active hosts under 100 later. Obviously, the BN-SA enables less active hosts than other 12 combinational algorithms, so it will reduce more energy consumption. Fig.3 shows the change of migrations along with time of consolidation. These algorithms all trigger many VM migrations in the former 10 periods and keep the number of migrations under 100 later. Because that these algorithms shut down a large number of hosts in early phase, a lot of VMs will be migrated to new destination hosts from those hosts. Moreover, in former period, the BN-SA shut down more hosts than other 12 combinational algorithms that lead to more migrations. But the BN-SA keeps the number of migrations around 70 which lower than other 12 combinational algorithms afterwards. Fig.4 shows the change of standard deviation of CPU utilization by active hosts which can reflect the state of load balance. Along with the increasing of sleep hosts, 12 combinational algorithms lead to load unbalanced. The BN-SA shows more unbalance than other 12 combinational algorithms in early phase, because that BN-SA keep less active hosts and higher density of VMs per host than them in former periods. But the standard deviation of hosts activated by BN-SA keep around 0.15 and is better than other 12 combinational algorithms afterwards.

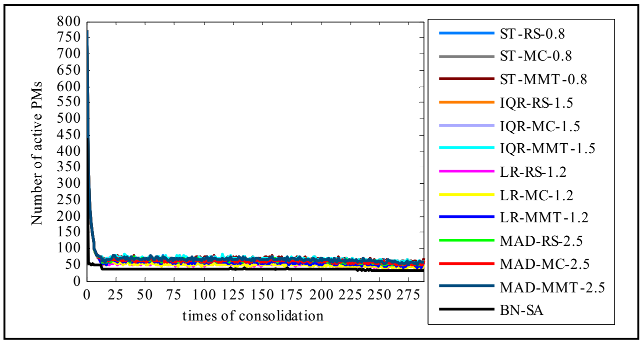

Fig. 2. The number of active hosts varies with time

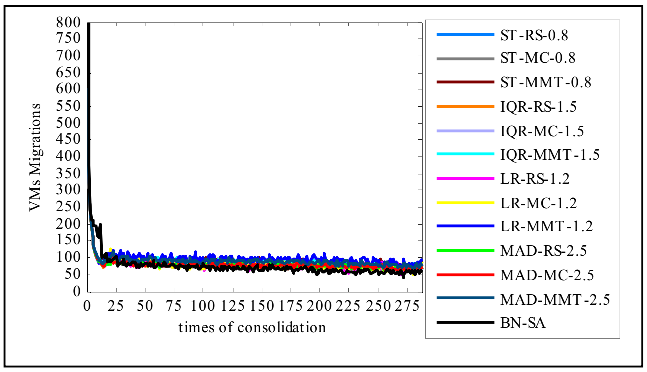

Fig. 3. The number of migrations varies with time

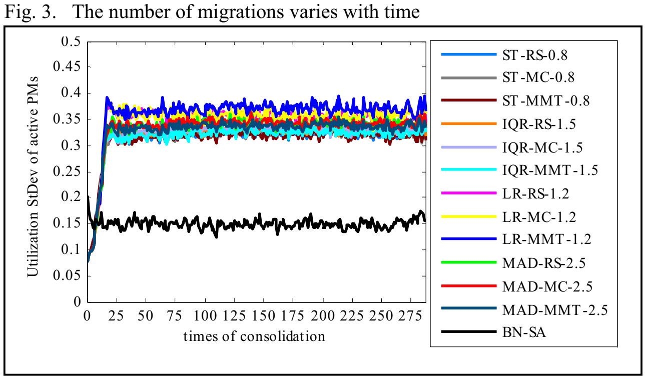

Fig. 4. The standard deviation of active host CPU utilization varies with time

V. CONCLUSION

This paper has studied the detail issue of selecting VMs to be migrated in VMs consolidation with consideration of the uncertain workload in data centers. Through building an estimation model depending on Bayesian Networks, BN-SA was proposed. BN-SA determines when a host is considered as being overloaded in terms of the overloaded probability of physical host. According to the conditional probability provided by Bayesian Networks, BN-SA selects particular VMs to migrate from overloaded hosts. Experimental results show that BN-SA can improve the QoS while decreasing energy consumption and reducing VM migrations. Here, we just only consider a few factors in data centers, further research can be processed in multiple resources.

REFERENCES

[1] Li. M, J.Bi and Z. Li, "Resource-scheduling-waiting-aware virtual machine consolidation," Journal of Software,vol.25, pp.1388-1402, 2014.

[2] R. Birke, L Y. Chen and E. Smirni, "Data Centers in the Cloud: A Large Scale Performance Study," Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, Honolulu, HI, pp. 336-343, 2012.

[3] A. Gandhi, M. Harchol-Balter, R. Das, et al, "Optimal power allocation in server farms," ACM SIGMETRICS Performance Evaluation Review, vol.37, no.1, pp.157-168, 2009.

[4] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," Proceedings of the 2008 conference on Power aware computing and systems, vol. 10, pp.1-5, 2008.

[5] A. Verma, P. Ahuja, and A. Neogi, "pMapper: power and migration cost aware application placement in virtualized systems," ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, Springer Berlin Heidelberg, pp.243-264, 2008.

[6] E. Feller, L. Rilling and C. Morin, "Energy-Aware Ant Colony Based Workload Placement in Clouds," 2011 IEEE/ACM 12th International Conference on Grid Computing, Lyon, pp. 26-33, 2011.

[7] A. Beloglazov, R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience, vol.24, no.13, pp.1397-1420, 2012.

[8] W. Voorsluys, et al, "Cost of virtual machine live migration in clouds: A performance evaluation," IEEE International Conference on Cloud Computing, Springer Berlin Heidelberg, pp.254-265, 2009.

[9] A. Beloglazov, J. Abawajy, R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," Future generation computer systems, vol.28, no.5, pp. 755-768, 2012.

[10] F. Farahnakian, T. Pahikkala, P. Liljeberg and J. Plosila, "Energy Aware Consolidation Algorithm Based on K-Nearest Neighbor Regression for Cloud Data Centers," Utility and Cloud Computing, 2013 IEEE/ACM 6th International Conference on, Dresden, pp. 256-259, 2013.

[11] S B. Shaw, A k. Singh, "Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center," Computers & Electrical Engineering, vol.47, pp.241-254, 2015.

[12] N. Fenton, M. Neil, "Making decisions: using Bayesian nets and MCDA," Knowledge-Based Systems, vol.14, no.7, pp.307-325, 2001.

[13] M. Wang, X. Meng and L. Zhang, "Consolidating virtual machines with dynamic bandwidth demand in data centers," INFOCOM, 2011 Proceedings IEEE, Shanghai, pp. 71-75, 2011.

[14] M. Chen, H. Zhang, Y. Y. Su, X. Wang, G. Jiang and K. Yoshihira, "Effective VM sizing in virtualized data centers," 12th IFIP/IEEE International Symposium on Integrated Network Management and Workshops, Dublin, pp. 594-601, 2011.

[15] R N. Calheiros, R. Ranjan, A. Beloglazov, et al, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software Practice & Experience, vol.41, no.1, pp.23–50, 2011.

[16] K S. Park, V S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," ACM SIGOPS Review, vol.40, no.1, pp.65-74, 2006.