Host load Prediction-based GMDH-EA and MMTP for Virtual Machines Load Balancing in Cloud Environment

Chenglei Peng, Qiangpeng Yang, Yao Yu, Yu Zhou, Ziqiang Wang, Sidan Du
School of Electronic Science and Engineering, Nanjing University, Nanjing, China

Abstract - Virtual machines (VMs) dynamic consolidation is effective to improve the utilization of resources and energy efficiency in cloud environment. However, the obligation of providing high quality of service to customers leads to the necessity in dealing with the energy performance trade-off; as aggressive consolidation may lead to performance degradation. Current solutions to the problem of host load detection are generally heuristic based. We propose a novel load balancing approach that combines the Group Method of Data Handling (GMDH) based on Evolutionary algorithm (EA) for host load prediction and the Minimum Migration Time policy (MMTP) for VMs migration. The GMDH-EA algorithm could predict the actual host load in each consecutive future time interval. We evaluate our method using the host load traces in the Google data centers with thousands of machines. The proposed algorithms significantly reduce energy consumption, while ensuring a high level of adherence to the Service Level Agreements (SLAs).

Keywords: Cloud Computing, Dynamic Consolidation, Host Load Prediction, Group Method of Data Handling, Minimum Migration Time

1 Introduction

The proliferation of Cloud computing has resulted in the consuming of enormous amounts of electrical energy. One of the ways to address the energy inefficiency is to leverage the capabilities of the virtualization technology [1]. The virtualization technology allows Cloud providers to create multiple VMs instances on a single physical server. And the reduction in energy consumption can also be achieved by switching idle hosts to low-power modes (i.e., sleep, hibernation), thus eliminating the idle power consumption.

However, efficient resource management in Clouds is not trivial, as modern service applications often experience highly variable workloads causing dynamic resource usage patterns. Therefore, aggressive consolidation of VMs can lead to performance degradation when an application encounters an increasing demand resulting in an unexpected rise of the resource usage. Ensuring reliable Quality of Service (QoS) defined via Service Level Agreements (SLAs) established between Cloud providers and their customers is essential for Cloud computing environments; therefore, Cloud providers have to deal with the energy-performance trade-off – the minimization of energy consumption, while meeting the SLAs.

The focus of this work is on energy and performance efficient resource management strategies that can be applied in a virtualized data center by a Cloud provider (e.g. Google App Engine). We investigate performance characteristics for the problem of energy and performance efficient dynamic VM consolidation considering multiple hosts and multiple VMs. Effective host load prediction is conducive to dynamic resource provisioning [2], virtual machine migration [3], server consolidation and energy management. Therefore, accurate host load prediction is essential for load balancing.

In this paper, we propose an effective host load prediction method with comparatively less prediction errors and acceptable prediction interval length. The main idea of our approach is to use GMDH-EA method based on evolutionary algorithm for host load prediction and apply Minimum Migration Time policy (MMTP) to the VM selection stage.

The GMDH method is a self organizing method first developed by Ivakhnenko [4] and it has been applied to solve many prediction problems with success. Zadeh et al. [5] proposed a new GMDH-type neural network where evolutionary algorithm is deployed to design the whole architecture of the network. We have combined the Phase Space Reconstruction (PSR) and GMDH for host load prediction in previous work [6]. The MMTP migrates a VM that requires the minimum time to complete a migration, firstly proposed by Beloglazov A. et al. [7].

We evaluate the proposed algorithms by extensive simulation using the Cloudsim toolkit and the cluster workload traces from 29 days of the resource usage by about 11k machines in Google data centers.

In this paper, we make the following contributions:

1. Our proposed method could predict the actual host load rather than the mean load only, the performance of our method has been investigated by different time intervals, i.e. 0.5h to 3h.

2. To the best of our knowledge, this is one of the first works to combine the GMDH-EA and MMTP approaches for host load balancing in the context of Cloud Computing.

3. An extensive simulation-based evaluation and performance analysis of the proposed algorithms.

The remainder of the paper is organized as follows. In Section 2 we discuss the related work. We present a thorough analysis of the VM consolidation problem in Sections 3. In
In contrast to the discussed studies, we combine the GMDH-EA and MMTP algorithms for dynamic adaption of VM allocation at run-time according to the current utilization of resources applying live migration, switching idle hosts to the sleep mode, and thus minimizing energy consumption. The proposed approach can effectively handle strict QoS requirements, multi-core CPU architectures, heterogeneous infrastructure and heterogeneous VMs.

According to the experiment results, our method achieves higher accuracy than the previous methods in mean load prediction. And what’s more, our method can predict the actual load variation with a lower MSE over a long time interval, which is very important to the VMs consolidation. While combined with MMTP, our algorithms can adapt the behavior according to the observed performance characteristics of VMs.

3 The VM consolidation problem

In this section we analyze the problem of dynamic VM consolidation considering multiple hosts and multiple VMs. VM consolidation is the key problem that IaaS provider or data center operators often face. They need develop appropriate resource management and scheduling strategies to meet SLAs, improve load balancing capability and reduce energy consumption. Before the VM selection stage, we need know which host is overloaded. Then the next step is to select particular VMs to migrate from this host.

We define that there are $n$ homogeneous hosts, and the capacity of each host is $A_h$. Although VMs experience variable workloads, the maximum CPU capacity that can be allocated to a VM is $A_p$. Therefore, the maximum number of VMs allocated to a host when they demand their maximum CPU capacity is $m = \frac{A_h}{A_p}$. The total number of VMs is $nm$. VMs can be migrated between hosts using live migration with a migration time $t_m$. Obviously, SLA violation occurs when the total demand for the CPU performance exceeds the available CPU capacity $A_h$. The cost of power is $C_p$, and the cost of SLA violation per unit of time is $C_v$. Without loss of generality, we can define $C_p = 1$ and $C_v = s$, where $s \in R^+$. We assume that when a host is idle, i.e., there are no allocated VMs, it is switched off and consumes no power, or switched to the sleep mode with negligible power consumption. We call non-idle hosts active. The total cost $C$ is defined as follows:

$$C = \sum_{t=t_0}^{T} (C_p \sum_{i=0}^{n} a_{it} + C_v \sum_{j=0}^{n} v_{jt})$$ (1)

Where $t_0$ is the initial time; $T$ is the total time; $a_{it} \in \{0,1\}$ indicating whether the host $i$ is active at the time $t$; $v_{jt} \in \{0,1\}$ indicating whether the host $j$ is experiencing an SLA violation at the time $t$. The problem is to determine when, which VMs and where should be migrated to minimize the total cost $C$. 

Section 4 we introduce the system model used in the development of heuristics for the dynamic VM consolidation problem. We propose our algorithms in Section 5, continuing with an evaluation and analysis of the obtained experiment results in Section 6. We discuss future research directions and conclude the paper in Section 7.

2 Related work

Many efforts [8][9][10] have been made in host load prediction in Grids or HPC systems. C. Dabrowski et al. [8] perform the host load prediction by leveraging the Markov model via a simulated environment. S. Akioka, et al. [9] combine the Markov model and seasonal analysis to predict the host load for one-step ahead in a computational Grid. Y. Wu et al. [10] use hybrid model for multi-step ahead host load prediction, which combines the Auto Regressive (AR) model and Kalman filter. Although the previous methods have achieved high accuracy for host load prediction in Grids, the Cloud host load holds a different scenario. Google’s traces show that the Cloud host load has more drastic fluctuation and higher noise, which we can see in [11].

B. Guenter [12] proposed a simple linear prediction scheme which predicts the host load for the next time. Q. Zhang [13] used the Auto-Regressive Integrated Moving Average (ARIMA) model to predict the host load. In [12], the ARIMA model could predict the load over a time window $H$ by iterated the one step prediction. In [14], D. Yang et al. proposed a multi-step-ahead prediction method for CPU load. Their method contains three consequent steps. The first step is to find a fit function for the change range sequence of the original sequence. The second step is to predict the multi-step-ahead change pattern. However, the length of the immediately preceding sequence that is used to find the same sequence and derive the change patterns from the history data is not discussed.

S. Di et al. [15] firstly use the Bayesian model to predict the host load in the Cloud. They proposed 9 novel features to characterize the recent load fluctuation in the evidence window, and could predict the mean load over consecutive time intervals. However, their method has two limitations. The first one is that the training period in evaluation type B is not discussed.

Sriikantaiah et al. [16] have studied the problem of request scheduling for multi-tier web applications in virtualized heterogeneous systems to minimize energy consumption, while meeting performance requirements. The authors have found that the energy consumption per transaction results in a “U”-shaped curve, and it is possible to determine the optimal utilization point. To handle the optimization over multiple resources, they proposed a heuristic for the multidimensional bin packing problem as an algorithm for the workload consolidation. However, the proposed approach is workload type and application dependent, whereas our algorithms are independent of the workload type, and thus are suitable for a generic Cloud environment.
4 The system model

In this paper, the targeted system is an IaaS environment, represented by a large-scale data center consisting of N heterogeneous physical hosts. Each host i is characterized by the CPU performance defined in MIPS, amount of RAM and network bandwidth. The storage is provided as an NAS to enable live migration of VMs. Multiple independent users submit requests for provisioning of M heterogeneous VMs characterized by requirements to processing power defined in MIPS, amount of RAM and network bandwidth. The fact that the VMs are managed by independent users implies that the resulting workload created due to combining multiple VMs on a single physical host is mixed. The mixed workload is formed by various types of applications which utilize the resources simultaneously. The users establish SLAs with the resource provider to formalize the QoS delivered. The provider pays a penalty to the users in cases of SLA violations.

The software layer of the system is tiered comprising local and global managers (Figure 1).

![Fig. 1. The tiered system model](image)

The local managers reside on each host as a module of the VM manager. Their objective is the continuous monitoring of the host’s CPU utilization, resizing the VMs according to their resource needs, and deciding when and which VMs should to be migrated from the host (3). The global manager resides on the master host and collects information from the local managers to maintain the overall view of the utilization of resources (1). The global manager issues commands for the optimization of the VM placement (2). VMsM perform actual resizing and migration of VMs as well as changes in power modes of the hosts (4).

Based upon the above model, we propose an new hybrid control system (Figure 2) whose core components include: host load analyzer, host load scheduler, and VM monitor.

![Fig. 2. The hybrid control system model](image)

The Analyzer analyzes the changes in the load, using GMDH-EA algorithm to predict the future host loads; the scheduler mainly focus on the integrated management and scheduling, according to the actual load, predicted load, state parameters and other information resources.

The most important feature of this control system is based on the hybrid control mechanisms by combination of active control of prediction and passive control of feedback. Through the hybrid control system, we can not only be informed of the fluctuations of host load in advance by the prediction technique so that can allows the scheduler to implement more calmly the VMs migration policies. Therefore, the system can target to advance to play a preventive role. But we can also be informed of the actual implementation of the scheduling policy through feedback technique so that can play a role in real-time corrective control action.

5 The algorithms for VM consolidation

In this section, we propose several algorithms for dynamic consolidation of VMs based on an analysis of historical data of the resource usage by VMs. We split the problem of dynamic VM consolidation into four parts: (1) determining when a host is considered as being overloaded to migrate of one or more VMs from this host; (2) determining when a host is considered as being under-loaded 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 of the VMs selected for migration from either the overloaded or under-loaded hosts.

5.1 Host load prediction

Beloglazov A. et al. [17] apply an approach based on the idea of setting fixed utilization thresholds. However, fixed utilization thresholds are not efficient for IaaS environments with mixed workloads that exhibit non-stationary resource usage patterns. Also they use Local regression algorithm first proposed by Cleveland [18]. The main idea of the method of local regression is fitting simple models to localized subsets of data to build up a curve that approximates the original data.

In this section, we propose the GMDH-EA method for the host load prediction.

5.1.1 The overview of GMDH-EA

The GMDH network is a feed-forward network that can be represented as a set of neurons, of which different pairs in each layer are connected through a quadratic polynomial and thereby produce new neurons in the next layer. The coefficients of the neuron are estimated using the Least Squares Method. The most popular base function used in GMDH is the gradually complicated Kolmogorov-Gabor polynomial:

$$\hat{y} = a_0 + \sum_{i=1}^{n} a_i x_i^2 + \sum_{j=1}^{n} \sum_{k=1}^{n} a_{ijk} x_i x_j x_k + \ldots$$

where $n$ is the number of the data in the dataset; $A = (a_0, a_1, a_2, \ldots)$ and $X = (x_1, x_2, x_3, \ldots)$ are the vectors of the
coefficients and input variables of the multi-input single-output system; and \( \hat{y} \) is the output of an individual host. However, in the GMDH algorithm, the infinite Kolmogorov-Gabor polynomial is estimated by a cascade of a second order polynomials using only pairs of variables in the form of

\[
y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2
\]  

(3)

The basic form of the GMDH algorithm has several limitations, e.g., each host can only have two input variables, and the neurons in each layer are only connected to the host in its adjacent layer. Therefore, we choose GMDH-EA to remove these restrictions, as each neuron in GMDH-EA can have a different number of input variables as well as a different order of polynomial.

5.1.2 The presentation of GMDH-EA network

The representation of the GMDH-EA network should contain the number of input variables for each neuron, the best type of polynomial for each neuron, and which input variables should be chosen for each neuron. Therefore, the chromosome for each individual should contain three subchromosomes. Each subchromosome in our algorithm is represented as a string of integer digits.

![The chromosome represents the GMDH-EA network](Image)

Fig. 3. The chromosome represents the GMDH-EA network

Figure 3 shows an example of a chromosome which represents an GMDH-EA network. This GMDH-EA network consists of three layers, and the neurons number of each layer are 3, 2 and 1. The number of input variables of each neuron ranges from 2 to 4, and the type of polynomials ranges from 1 to 3.

5.1.3 Estimate the coefficients of each neuron

In the GMDH-EA network, the coefficients of each neuron are derived by minimizing the mean squared error between \( y \) and \( \hat{y} \).

\[
e = \frac{1}{Nt} \sum_{i=1}^{Nt} ||y_i - \hat{y}_i||^2
\]  

(4)

Where \( Nt \) is the size of the training set, and \( y_i \) and \( \hat{y}_i \) are the vectors of the actual and predict values. Using the training set, this gives rise to the set of linear equations

\[XC = Y\]  

(5)

The coefficients of each neuron are derived in the form

\[
(X^TX)^{-1}X^TY
\]  

(6)

Where \( Y = [y_1, y_2, ..., y_{Nt}]^T \), and the values of \( X \) and \( C \) are according to the number of input variables and the order of the polynomial.

5.1.4 The fitness function

The fitness function is very important to the GMDH-EA network, as it determines the performance of the model. In this paper, we use the locally weighted mean square error as the fitness function.

\[
\Phi = \frac{1}{Nv} \sum_{i=1}^{Nv} W_i ||y_i - \hat{y}_i||^2
\]  

(7)

Where \( Nv \) is the size of the validation set, \( W \) is the weighting function. There are many weighting functions proposed by the researchers [15]. In this paper, we use the tricube kernel weighting function as follows:

\[
W_i = (1 - \left( \frac{D_i}{\sum_j D_j} \right)^3)^3 \]  

(8)

Where \( D_i \) is the Euclidean distance of the input variables between the data in the validation set and the prediction set, which is used to indicate the similarity between the load in the validation set and the prediction set.

5.1.5 Crossover and mutation operations

The crossover and mutation operation are used to produce offsprings from two parents, which are chosen using the roulette wheel selection method. The crossover operation for the first and the second subchromosome is simply accomplished by exchanging the tail of each two subchromosomes from a random point. The change of the third subchromosome follows the change in the first one. The mutation operation is similar to the crossover operation.

5.2 VM selection

Once the system get the predicted load, it has been decided which host is overloaded or under-loaded. So the next step is to select particular VMs to migrate from this host. In this section we propose two policies for VM selection. The described policies are applied iteratively. After a selection of a VM to migrate, the host is checked again for being overloaded. If it is still considered as being overloaded, the VM selection policy is applied again to select another VM to migrate from the host. This is repeated until the host load is considered as being at the normal value.

5.2.1 The minimum migration time policy

The Minimum Migration Time policy (MMTP) migrates a VM \( v \) that requires the minimum time to complete a migration relatively to the other VMs allocated to the host. The migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host \( j \). Let \( V_j \) be a set of VMs currently allocated to the host \( j \). The MMT policy finds a VM \( v \) that satisfies conditions formalized in (9).

\[
v \in V_j \forall a \in V_j \frac{RAM_v(a)}{NET_j} \leq \frac{RAM_a(a)}{NET_j}
\]  

(9)

Where \( RAM_v(a) \) is the amount of RAM currently utilized by the VM \( a \); and \( NET_j \) is the spare network bandwidth available for the host \( j \).

5.2.2 The Random Selection Policy (RSP)

The Random Selection Policy (RSP) selects a VM to be migrated according to a uniformly distributed discrete random variable \( X \equiv U(0, |V_j|) \), whose values index a set of VMs \( V_j \) allocated to a host \( j \).
5.3 VM placement

The VM placement can be seen as a bin packing problem with variable bin sizes and prices, where bins represent the physical hosts; items are the VMs that have to be allocated; bin sizes are the available CPU capacities of the hosts; and prices correspond to the power consumption by the hosts. As the bin packing problem is NP-hard, to solve it we choose a modification of the BFD algorithm denoted Power Aware Best Fit Decreasing (PABFD) proposed by Beloglazov A. et al. [6], we sort all the VMs in the decreasing order of their current CPU utilizations and allocate each VM to a host that provides the least increase of the power consumption caused by the allocation. The pseudo code for the algorithm is presented in Algorithm 1. The complexity of the algorithm is \( nm \), where \( n \) is the number of hosts and \( m \) is the number of VMs that have to be allocated.

\[
\text{Algorithm 1: Power Aware Best Fit Decreasing (PABFD)}
\]

1Input: hostLst, vmLst Output: allocation of VMs
2vmLst.sortDecreasingUtilization()
3foreach vm in vmLst do
  4  minPower ← MAX
  5  allocatedHost ← NULL
  6  foreach host in hostLst do
  7    if host has enough resources for vm then
  8      power ← estimatePower(host, vm)
  9      if power < minPower then
 10        minPower ← power
 11        allocatedHost ← host
 12      if allocatedHost ≠ NULL then
 13        allocation.add(vm, allocatedHost)
14return allocation

6 Performance evaluation

6.1 Experiment setup

The CloudSim toolkit [19] has been chosen as a simulation platform, as it is a modern simulation framework aimed at Cloud computing environments. It has been extended to enable energy-aware simulations, as the core framework does not provide this capability. Apart from the energy consumption modeling and accounting, the ability to simulate service applications with dynamic workloads has been incorporated.

We have simulated a data center that comprises 1000 heterogeneous hosts. The number of VMs is 1600. All the load traces are real data coming from the 29 days of the resource usage by about 11k machines in Google datacenters [20]. We have randomly chosen 10 times load traces to form 10 data centers. For host load prediction method, we evaluated GMDH-EA and gave the actual prediction in different time intervals, i.e. 0.5h to 3h. The GMDH-EA parameters are shown in Table I, which are optimized to get the best performance for the load prediction.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max generation</td>
<td>50</td>
</tr>
<tr>
<td>Population size</td>
<td>35</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.9</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.15</td>
</tr>
<tr>
<td>Number of layers</td>
<td>4</td>
</tr>
<tr>
<td>Number of neurons of each layer</td>
<td>9,6,3,1</td>
</tr>
<tr>
<td>Number of inputs to be selected</td>
<td>2-4</td>
</tr>
<tr>
<td>Polynomial type</td>
<td>1-3</td>
</tr>
</tbody>
</table>

6.2 Test metrics

In order to compare the efficiency of the algorithms we use several metrics to evaluate their performance. One of the metrics is the total energy consumption (EC) by the physical servers of a data center caused by the application workloads. The second metric is the level of SLA violations (SLAV). Another metric is the number of VM migrations initiated by the VM manager during the adaptation of the VM placement. All these three metrics results can be found in the CloudSim output.

6.3 Host load prediction

The accurate prediction of host load in a Cloud computing data center is very important to improve resource utilization, lower data center costs and ensure the job performance. The previous methods [12][13] for multi-step ahead prediction usually iterate the result of the one-step ahead prediction, which will generate cumulative errors.

However, the output of our proposed method is a vector of the host load, which will not generate cumulative errors regardless of the step length, as the current predict value has nothing to do with the last predict value. We quantified the performance of actual load prediction with mean squared error (MSE).

\[
MSE = \frac{1}{H} \sum_{i=1}^{H} (A_i - F_i)^2
\]  

(10)

Where \( H \) is the step length, \( A_i \) and \( F_i \) are the actual value and forecast value.

![Fig. 4. MSE of actual load prediction](image)
In Figure 4, we compare our method with the AR method and the Pattern Prediction (PP) method proposed by Yang [14]. The average MSE of our method in 3h ahead prediction is 0.0046, which is much lower than the other two methods. What’s more, we can find that our proposed method keeps a good performance with the prediction step increases, while the performance of the other two methods has a large degree of decline.

Fig. 5. Actual load prediction.

Figure 5 shows the load prediction results of hosts in the Google data center. As the interval in Google trace is 5 min, the step length of 0.5h to 3h is 6 to 36. And the y-label in Figure represents the CPU utilization, which has been normalized.

Our prediction result shows that the proposed method could achieve high accuracy although the host load fluctuates more drastically. As we can see in Figure 6, 7 and 8, our proposed method can still get a satisfactory performance.

6.4 Simulation results

To make a simulation-based evaluation applicable, it is important to conduct experiments using workload traces from a real system. For our experiments we have used data coming from the cluster workload traces of Google datacenters. The interval of utilization measurements is 5 minutes. We have randomly chosen record of 1600 tasks running on 1000 hosts of 29 days from the workload traces collected from May 2011 [20]. During the simulations, each VM is randomly assigned a workload trace from one of the VMs from the corresponding day. In the simulations we do not limit the VM consolidation by the memory bounds, as this would constrain the consolidation, whereas the objective of the experiments is to stress the consolidation algorithms.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Energy (KWH)</th>
<th>SLA Violation (%)</th>
<th>VM migration ( \times 10^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR-RSP</td>
<td>84.94</td>
<td>4.38</td>
<td>17.98</td>
</tr>
<tr>
<td>LR-MMTP</td>
<td>83.82</td>
<td>4.32</td>
<td>17.33</td>
</tr>
<tr>
<td>GMDH-RSP</td>
<td>83.54</td>
<td>4.30</td>
<td>16.77</td>
</tr>
<tr>
<td>GMDH-MMTP</td>
<td>81.93</td>
<td>4.26</td>
<td>13.37</td>
</tr>
</tbody>
</table>

The average results of 10 data centers of the combinations of each host load detection algorithm and the MMT policy are shown in Table II.

We have simulated all combinations of the host load detection algorithms (LR and GMDH) and VM selection policies (MMTP and RSP). The results produced by the selected algorithms are shown in Figure 6, 7 and 8.

From the observed simulation results, we can make several conclusions: (1) the GMDH-EA algorithm outperforms the local regression algorithm; (2) the MMTP policy produced better results compared to the RSP policy, meaning that the minimization of the VM migration time is more important;(3) the combination of GMDH-EA with MMTP algorithms outperform others.

7 Conclusion

To maximize ROI, Cloud providers have to apply energy-efficient resource management strategies, such as dynamic VMs consolidation and switching idle servers to power-saving modes. However, such load balancing is not trivial, as it can result in the SLA violations. In this paper we
have conducted competitive analysis of the VM load balancing problems.

We proposed to combine GMDH-EA and MMTP algorithms for optimal online deterministic algorithms for these problems. According to the results of the analysis, we have proposed novel adaptive heuristics that are based on an analysis of historical data. We have also evaluated the proposed algorithms through extensive simulations on a large-scale experiment setup using workload traces from more than 11k machines in Google data centers. The results of the experiments have shown that the proposed GMDH-EA prediction algorithm combined with the MMTP selection policy significantly outperforms other VM consolidation algorithms in regard to the MSE metric due to a lower value in a long time interval and a substantially reduced level of SLA violations and the number of VM migrations.

In order to evaluate the proposed algorithm in a real Cloud environment, we plan to implement it by extending a real-world Cloud platform with commercial partner. Besides the reduction in infrastructure and on-going operating costs, this work also has social significance as it will decrease the carbon dioxide footprints and energy consumption by modern IT infrastructures.

8 Acknowledgment
This work was supported by the scientific and technical supporting programs of Jiangsu province, china, No.BE2011169.

9 References