A Resource Reduction Scheme with Low Migration Frequency for Virtual Machines on a Cloud Cluster

Changhyeon Kim[1], Wonjoo Lee[2], and Changho Jeon[1]
[1]Department of Science and Computer Engineering, Hanyang University(ERICA Campus),
Ansan, 426-791, Republic of Korea
[e-mail: {ctcquatre, chj5193}@hanyang.ac.kr]
[2]Department of Computer Science, Inha Technical College,
Incheon, 402-752,Republic of Korea
[e-mail: wonjoo2@inhatc.ac.kr]
*Corresponding author: Changho Jeon

Received December 26, 2012; revised May 2, 2013; accepted May 18, 2013; published June 26, 2013

Abstract

A method is proposed to reduce excess resources from a virtual machine(VM) while avoiding subsequent migrations for a computer cluster that provides cloud service. The proposed scheme cuts down on the resources of a VM based on the probability that migration may occur after a reduction. First, it finds a VM that can be scaled down by analyzing the history of the resource usage. Then, the migration probability is calculated as a function of the VM resource usage trend and the trend error. Finally, the amount of resources needed to eliminate from an underutilized VM is determined such that the migration probability after the resource reduction is less than or equal to an acceptable migration probability. The acceptable migration probability, to be set by the cloud service provider, is a criterion to assign a weight to the resource reduction either to prevent VM migrations or to enhance VM utilization.

The results of simulation show that the proposed scheme lowers migration frequency by 31.6~60.8% depending on the consistency of resource demand while losing VM utilization by 9.1~21.5% compared to other known approaches, such as the static and the prediction-based methods. It is also verified that the proposed scheme extends the elapsed time before the first occurrence of migration after resource reduction 1.1~2.3-fold. In addition, changes in migration frequency and VM utilization are analyzed with varying acceptable migration probabilities and the consistency of resource demand patterns. It is expected that the analysis results can help service providers choose a right value of the acceptable migration probability under various environments having different migration costs and operational costs.

Keywords: Virtual machine migration, resource reduction, migration frequency, resource usage trend, migration probability

This work was supported by the research fund of College of Engineering Science at Hanyang university (HY-2011-G)

http://dx.doi.org/10.3837/tiis.2013.06.003

1. Introduction

Cloud computing has recently become a popular computing paradigm which is characterized by virtualized resources provided on a per-demand basis. Virtualization is a key technique for cloud computing and allows virtual machines(VMs) to run in an independent operating environment without being restricted by physical resource boundaries. Several VMs can be hosted on a single physical machine(PM) as long as the total amount of resources allocated to them does not exceed the resource capability of the PM. A VM that runs short of resources can migrate to another PM that can provide the required resources. These features allow decoupling of applications and service from the physical servers and provide scalability and flexibility in resource management to enable the applications to use just the right amount of resources for their service [1]. Thus, virtualization is an attractive alternative to reduce the operational costs of IT organizations that maintain operating servers for a peak workload rather than for the average workload [2]. For this reason many IT organizations are providing cloud service or are carrying out related projects, such as Amazon EC2, IBM Blue Cloud.

In general, a cloud cluster is a large-scale computing center and organized with thousands of machines. Therefore, it consumes a huge amount of energy for operation, which is a critical issue among all IT organizations. A report from National Resources Defense Council [3] pointed out that an idle server uses 69-97% of the total energy consumption, even when the power management function is working. Also, an idle server consumes almost half of the energy used by a fully loaded server [4]. The main purpose of parallel computing systems is to maximize system throughput, with little concern about the energy consumption. In a large scale cloud computing environment, however, energy consumption is a serious problem to solve.

A viable strategy for reducing the energy consumption of a cloud cluster is server consolidation which vacates some servers through VM placement and turns off their power to save energy [5]. Since the amount of VM resources is used as an input to the placement algorithm, the estimation of the resource amount that will be allocated to a VM is a key factor that affects the performance of the consolidation. If the resource allocated to a VM is more or less than the resource requirement of the applications running on it, the VM becomes underutilized or overutilized, respectively, after server consolidation. Underutilized VMs might diminish the advantage of energy saving from the server consolidation. On the other hand, overutilized VMs would require migrations due to the resource shortage, thus incurring migration overhead. This also diminishes the benefit of server consolidation.

Migration overhead includes increased network traffic, excessive disk I/O, and the consumption of CPU time caused during the course of transferring memory pages and data related to the operation of the source and destination VMs. The amount of increased network traffic and the disk I/O depends on the size of memory, the volume of the data, and their update frequency. Migration overhead decreases the throughput of the network and disk of the PMs to which the source and destination VMs belong, resulting in degradation of the performance of the other VMs hosted on the two PMs [6]. After the completion of the transmission, the VM is temporally suspended to update the dirty page and the address resolution protocol(ARP) table entry. The suspension duration is called the downtime. As the applications hosted on the VM freeze during the downtime, migrations can lead to service disruptions and increases in the service response time [7]. Therefore, the amount of resources that will be allocated to a VM should be deliberately determined so that migrations can be avoided.

When considering both VM utilization and the migration overhead together, it would be ideal to assign the amount of resources necessary for applications on a VM; however, application resource demand may change over time. For that reason, the resources, once assigned, cannot meet application demand continuously. That is, a VM can be over-utilized or under-utilized according to circumstance. This paper deals with the problem of enhancing VM utilization by reducing redundant resources on runtime basis when a VM is under-utilized. The focus of our proposal is to reduce resources so that subsequent migrations are avoided. The proposed scheme predicts the resource demand by analyzing the resource usage history of a VM and then estimates the probability of VM migration for the amount of resource reduction to be achieved. The amount of reducible resources is determined based on the estimated migration probability. Since there is a positive correlation between the amount of resources to be reduced from a VM and the probability of VM migration after the reduction, a parameter called the acceptable migration probability is introduced in order to balance the migration frequency against the VM utilization on resource reduction.

This paper is organized as follows. Section 2 reviews related works. Section 3 presents our resource reduction algorithm. In Section 4 the performance of the proposed scheme is analyzed in comparison with other approaches through simulation. A pattern generator that is implemented to supply various resource demand patterns for the simulation is described, and the results of the simulation are discussed in detail. Finally, Section 5 concludes the paper.

2. Related Works

Although VM migration is a practical means to provide flexibility, load balancing, and fault tolerance from the viewpoint of resource management, its overhead has not been seriously considered. However, several recent studies have pointed out that migration overhead is not negligible. Lim et al. [6] showed that the completion time of a job executed by VMs running on a pair of PMs increases due to the network bandwidth consumption and intensive disk I/O caused by the migration. Zhao et al. [7] reported performance degradation that resulted from the CPU and memory-intensive workload when many VMs migrate simultaneously. Voorsluys et al. [8] verified an increase in the application response time and the incurrence of downtime during the migration process. They also noted that the response time rapidly increases upon completion of migrations in order to handle the requests that arrived during the downtime, which may lead to a violation of service level agreement(SLA). Clark et al. [9] suggested a writable working set, which is a set of frequently updated memory pages, to reduce the migration time. The idea is to avoid unnecessary transfers and, hence, to reduce the total migration time by transmitting this set at the final stage of migration.

Several studies on server consolidation have considered migration overhead as a constraint for VM placement. Hermenier et al. [10] pointed out that the migration overhead can offset the benefit of server consolidation and presented a dynamic algorithm for reducing the migration overhead in VM placement. Ho et al. [11] proposed a server consolidation algorithm with a bounded cost of VM relocation. They analyzed the relocation cost including the runtime overhead and extra energy consumption due to the migration, and found a theoretical bound on the relocation cost that assures consolidation quality. Jung et al. [12] pointed out that the migration overhead can be an obstacle to guaranteeing the response-time-based SLA and found a VM configuration using a prediction model and graph search techniques. Verma et al. [13] proposed an algorithm using a migration cost function that can reduce the migration and power costs. Beloglazov et al. [14] proposed a VM placement algorithm for minimizing the number of migrated VMs by migrating to a PM with lower CPU utilization only if the upper utilization threshold is violated. All of these server consolidation algorithms have a common approach to the VM placement problem that increases the utilization of the PMs and reduces the power cost. In this approach, the utilization of a PM can be further improved by increasing the utilization of the VMs.

The utilization of the VMs can be improved if unnecessary resources are reduced. However, it is not easy to estimate the amount of unnecessary resources in advance because the resource demand changes dynamically. Prediction is often used to estimate the resource demand of applications as reported in several works. Wood et al. [15] predicted the resource demand of an application using a regression-based model. Their model, which is derived from the resource usage profiles of a PM and the applications, estimates the resource requirements of VMs to be virtualized on a given platform. Wood et al. [16] designed a resource management system that detects hotspots and mapped the physical resources to the virtual resources. The system predicts the resource demand through autoregression, using a resource profile obtained from the resource usage history, and estimates the future peak needs from a high-percentile distribution profile. Gong et al. [17] presented a Markov chain model to predict the resource demand. In this model, each state contains a resource demand range, and the future resource demand is found from the transitions between states. Ganapathi et al. [18] predicted the resource demand through Kernel canonical correlation analysis. It turns out that prediction technique is commonly used to estimate the resource demand, but the accuracy of prediction based on the usage history tends to decrease when the time series is non-stationary. Thus, an error can occur between the predicted demands and the real demands, and such an error will lead to either of the following two situations. If a VM is supplied with more resource than its demand, it will be underutilized; otherwise, the VM will migrate to resolve the resource shortage.

Numerous studies of server consolidation have described attempts to reduce the power cost by enhancing the PM utilization in general. To enhance the PM utilization, the VM utilization first needs to be increased, which can be achieved by reducing any redundant resource. This, however, can cause VM migrations in an environment in which the resource demands change dynamically. If a VM is being fully utilized after downsizing, migration would be inevitable when the resource demand abruptly increases, which would be the case with a flash crowd. Therefore, considering the migration overhead and its side effects, the amount of resources to be reduced should be carefully determined in order to avoid migrations.

3. Resource Reduction Algorithm

The objective of our algorithm is to reduce the resources allocated to VMs while avoiding subsequent VM migrations. In order to avoid migration, a VM should have the amount of resource needed by the applications. If the applications running on a VM show consistent resource demands, the amount of VM resources can be determined in advance. Since the resource demands of applications tend to change dynamically, the amount of VM resources should be determined on a runtime basis.

Our proposed scheme is a runtime method for reducing the amount of VM resources based on the migration probability. The migration probability is defined as the likelihood that a migration will take place due to a shortage of resources. In our scheme, migration probability is estimated using the trend error, which is the difference between the measured value of resource usage and the value of the resource usage trend at a certain time instance. The resource usage trend is the profile of resource usage of a regularly-observed VM. A high trend error value means that the resource usage of a VM is sharply fluctuating, indicating a high migration probability. Conversely, a small trend error means that the resource usage of the VM is consistent in general, suggesting a low migration probability. Therefore, the estimation of the migration probability as a function of the trend error is the primary concern of this study. The amount of resources reduced is determined depending on the estimated migration probability.

3.1 Deciding Whether a VM can be Downsized

In order to decide whether a VM can be downsized it is necessary to anticipate how the resource usage will change in the future. Therefore, the VM resource usage trend needs to be found from consecutively-measured resource usage over a certain time period. The time period during which the trend is calculated is called the time slot. We obtain the resource usage trend with a simple linear-regression analysis of the resource usage. The trend is calculated at regular time intervals to reflect the most recent resource usage.

We use two types of resource usage trends: short-term and long-term. The former is obtained from a single time slot while the latter is obtained from a sequence of time slots. The long-term trend, however, is calculated from only a few recent time slots. The reasons for this are two-fold: a) because the resource usage measured over a longer time period is larger in volume and causes a heavy computation load, and b) because the resource usage measured longer ago is less dependable for trend calculation than recently measured usage. The short-term trend shows momentary changes in the resource usage during only one time slot. In contrast, the long-term trend demonstrates the general shape of changes in the resource usage, with little fluctuation over a relatively long time span.

The resource usage trend can be ascending, descending, or steady. If the long-term resource usage trend is descending or steady, the possibility that the resource usage during the next time slot will decrease or stay at the same level is high. In other words, it is very likely that the resource amount required in a time slot will be less than or equal to the resource amount allocated during the preceding time slots. However, even if the long-term resource usage trend of a VM appears to be descending or steady over a number of time slots, certain individual time slots among them may show an ascending trend. This can be caused by a trend changeover or irregular resource use, and if that is the case, the VM should not be considered as a candidate for resource reduction. Therefore, we cannot determine whether a VM can be downsized simply with the long-term trend because trend changeover or irregular resource usage cannot be detected.

In order to identify a trend changeover or irregular resource usage the short-term trend must be examined along with the long-term trend. By the nature of each trend, the difference in the values of the two trends implies that there is a variation in resource usage. The difference will likely grow as the resource usage fluctuates more widely. It will also increase after a trend changeover, if such an event occurs. The difference may result from distortion of the short-term trend due to long measurement intervals. Even if the short-term trend is distorted, however, the distortion range will be confined within the vicinity of the long-term trend owing to the nature of the linear regression.

To conclude, a VM can be downsized only if the deviation between the short- and long-term trends is within a certain range, given that both trends are descending. If the deviation exceeds this range, it could be an indication of a trend changeover or of irregular resource usage and, thus, the VM should not be downsized in such a situation. The acceptable range of deviation is a parameter of our proposed scheme to be set by the cloud service provider.

3.2 Calculating the Migration Probability

A VM may undergo a migration if the resource demand increases after resource reduction. Thus, a certain amount of extra resources needs to be set aside when reducing resources from a VM to avoid subsequent migrations. The amount of extra resources can be predicted from the resource usage profile of a VM. However, since the resource usage of VMs is not generally consistent, it is difficult to predict the correct amount of additional resources. An insufficient prediction would result in the extra resources provided being exhausted, causing a migration. Therefore, we first find the probability that the extra resources are exhausted, which is essentially the same as the migration probability defined in the beginning of Section 3.

The migration probability may vary depending on the application's tendency to use resources and the actual resource usage at the time of reduction. For example, consider an application whose resource usage fluctuates in a wide range, but, at the same time, tends to return to its average usage. This application will show a sharply ascending resource usage trend when its resource usage runs below this average. In contrast, it will show a weakly ascending resource usage trend when its resource usage exceeds this average. This suggests that a VM with a given amount of extra resources has a higher migration probability when its resource usage is below average than when its resource usage is above average. Applications that have frequent flash crowds generally show a sharply ascending resource usage trend until their resource usage reaches its peak. Therefore, if a VM is allocated fewer resources than its peak usage, it will have a high migration probability. On the other hand, a VM running applications that show a narrow variation range or a gradual increase in their resource usage has a relatively low migration probability compared to the two cases just mentioned, with a slight variation depending on the resource usage pattern of the VM. Therefore, in order to account for these attributes when calculating the migration probability, we first calculate the ascending resource usage trends for different ranges of resource usage. A range of resource usage is set by evenly dividing the maximum amount of resources that can be allocated to a VM. The ascending trend of a certain resource usage range enables us to predict an increase in resource usage of the same range.

The granularity of the range is controlled by the cloud service provider. With a finer granularity, the ascending resource usage trend can be closer to the given resource usage. However, an excessively fine granularity is not a good option because the ascending resource usage trends obtained from neighboring ranges too close together would become too similar to show a difference. In our scheme, the ascending resource usage trend is calculated from the resource usages during one time slot. Therefore, we recommend that the granularity of the range be set to the average of the resource increments during every time slot. The ascending resource usage trend for a resource usage range is found with the exponential average of the ascending trends shown so far with the resource usage in the range. The standard deviation of the trend error for a resource usage range is also obtained with the same function. Let us denote the ascending resource usage trend of the *i*-th range by a_i and the standard deviation of the trend error by σ_i.

Given the VM resource usage, the condition for migration can be defined by using the ascending resource usage trend of the corresponding resource usage range. If u, the resource usage measured last in a time slot, falls in the *i*-th range, the ascending resource usage trend of the next time slot will be most similar to a_i. Therefore, we assume that the ascending resource usage trend of the next time slot is the same as a_i. Then, given v as the amount of resources allocated to a VM, the predicted amount of redundant resources at time τ is $v - a_i\tau - u$. Then, with ε denoting the error of the ascending resource usage trend, the condition for migration occurring due to the exhaustion of the redundant resources is as follows:

$$v - a_i\tau - u < \varepsilon.$$

Now, let ε_{LB} be the lower limit of ε that leads to VM migration at time τ. Then, we can say that a migration will occur if ε is greater than or equal to ε_{LB}. Note that ε in the above inequality is an error term to be treated as a random variable in regression models. In studies dealing with the prediction of application workloads, the workload is predicted through the regression analysis, and the error term for the prediction is assumed to follow the Gaussian distribution[19][20]. Since our error term functions in the same way as theirs, we also assume that ε for a_i show a zero mean Gaussian distribution with the standard deviation σ_i. Then, the estimated migration probability with ε_{LB}, to be denoted by P_E, is calculated as follows:

$$P_E = \int_{\varepsilon_{LB}}^{\infty} \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\varepsilon^2/2\sigma_i^2}\, d\varepsilon.$$

3.3 Finding the Amount of Reducible Resources

The idea of our scheme is to determine the amount by which to reduce the VM resources in order to prevent migrations. To avoid migrations, one may want to allocate as many extra resources to a VM as possible. However, this is inefficient from the viewpoint of VM utilization. Therefore, we introduce a new parameter, called the acceptable migration probability, which serves as the criterion on resource reduction to give weight either to the reduction of the migration frequency or to the enhancement of the VM utilization. The acceptable migration probability, to be denoted by P_A herein, is also set by the cloud service provider.

The amount of resources to eliminate is determined such that migration probability at all instances of τ in the time slot after reduction does not exceed P_A. When the resource usage trend of a time slot is ascending, the migration probability becomes the highest at the last instance of τ in that time slot in general. We can say that, if that highest migration probability is smaller than P_A, the migration probability at all other time instances in the same time slot must be also lower than P_A. Therefore, we can cut resources down to the minimum resource allocation satisfying $P_A \ge P_E$ at the last time instance of the time slot. Thus, the amount of resources to be eliminated from a VM can be determined from the difference between the current resource allocation and such a minimum resource allocation.

Note that P_E is found as a function of the ascending resource usage trend of a VM and its error as described in the previous section. We assumed that the ascending resource usage trend within each range was equal to the exponential average of the ascending trends in the same range. In addition, the distribution of the trend error was also assumed to follow a Gaussian distribution.

However, there are two cases in which our assumptions may not hold. The first is when the ascending resource usage trend of individual ranges fluctuates sharply from time slot to time slot. The second case occurs when the error of the ascending resource usage trend does not follow a Gaussian distribution. Since trend errors imply deviation of the resource usage from the resource usage trend, they may not show a Gaussian distribution. In such cases P_E can be different from the true probability of migration occurrence. Let us denote the true migration probability in those two cases by P_{True}. If P_E is estimated to be higher than or equal to P_{True}, the migration probability after the resource reduction will not exceed P_A; otherwise, it will exceed P_A. Our objective is to reduce resources while maintaining a migration probability lower than or equal to P_A after the reduction. Therefore, we should find the amount of reducible resources in an additional step when P_E is smaller than P_{True}. In other words, we need to adjust P_E to make it as close to P_{True} as possible.

For such adjustment, we need to examine the migration occurrence ratio, that is, the ratio of the number of resource reductions that caused a migration in the first time slot after the resource reduction to the total number of resource reductions made. If the migration occurrence ratio is higher than P_A, it means that P_E has been underestimated. Therefore, we adjust P_E by adding the difference between the migration occurrence ratio and P_A, making the value close to P_{True}. In this case, we can cut the resources down to the minimum resource allocation satisfying $P_A \geq P_E + R_M - P_A$ at the last time instance of the time slot, where R_M is the migration occurrence ratio. As a consequence, the amount of resources to be eliminated is determined from the difference between the current resource allocation and such a minimum resource allocation. Note that the criterion for the minimum resource allocation here is different from that for the normal case, where the ascending resource usage trend in each range is equal to the exponential average of the ascending trends in the same range with Gaussian distribution of the trend error.

4. Performance Evaluation

The approaches to resource allocation used in studies on consolidation are divided into two types: static methods and predictive approaches. Static methods determine the amount of resources to allocate to a VM by comparing the amount of resources already allocated with the amount of resources being used. Predictive approaches determine the amount of resources to allocate by predicting the resources that will be needed in the near future. In this section, the performance of our proposed scheme is evaluated through simulation and compared with two known approaches: a static method and a predictive approach.

Since resource reduction is the main interest of our work, the comparison between different approaches focuses on the efficiency of resource reduction. The metrics used for this evaluation are VM utilization, time of migration occurrence, migration count, and the number of active PMs. All of these metrics can be measured in simulation with the workload of applications and the amount of resources allocated to VMs given. Please note that the costs which may be incurred in the process of VM downsizing, such as the degradation of I/O bandwidth, the increase in network traffic, the overhead for ARP table updates, the downtime, etc., are not evaluated. This is because VM downsizing cannot take place before decisions on resource reduction are made. That is, the evaluation of our resource reduction scheme does not depend on such costs.

The VMs and PMs are implemented to function as bin objects and bin containers, respectively, in our simulator written in C^{++}. The validity of the simulation depends on the resource demand pattern of the applications running on VMs. Several studies[21][22][23] have used trace data obtained from real servers. However, since there are limitations in collecting real trace data, it is difficult to cover a wide variety of resource demand patterns of applications. Therefore, we have designed and implemented a demand pattern generator in order to provide various resource demand patterns for use in the evaluation.

4.1 Resource Demand Pattern Generator

Applications that are executed in a cloud environment may show large fluctuations in resource demand. Even a single application might fluctuate substantially depending on the field of application services. Therefore, our resource demand pattern generator has been implemented with a consistency factor to assign a variety component to the demand fluctuation. The consistency refers to the degree of proximity between the resource demand values at two consecutive time instances.

The pattern generator is organized with the trend process and the residual process. The trend process assigns resource demand values to the trend points according to a given consistency, which forms the frame for a resource demand pattern. The residual process completes a resource demand pattern by adding residuals onto the line that connects the trend points.

Fig. 1 gives an outline of the trend process. d denotes the degree of consistency and ranges from 0 to 1. TP_i and n_{TP} are the i-th trend point and the number of trend points, respectively. avg is the average of the resource demand values given to all trend points, and f is the widest variation in the resource demand among all pairs of trend points. V is the pool of resource demand values, which ranges from $avg - f$ to $avg + f$.

The demand value at TP_i is determined in one of two ways depending on d . When d is less than $\text{Uniform}(0,1)$, it is determined by the Trend function. The function Trend returns an index of the demand value that is randomly chosen from V on the condition that the difference between the demand values at TP_{i-1} and TP_i is smaller than $(1-d)f$, and then it removes the demand value from V . On the contrary, when d is greater than or equal to $\text{Uniform}(0,1)$, the demand value at TP_i is determined with the value in V that is closest to that of TP_{i-1} . As a result, an increased value of d makes $(1-d)f$ smaller and is more likely to become greater than or equal to $\text{Uniform}(0,1)$, thereby, allowing smoother demand patterns to be generated.

```
for $i = 1$ to $n_{TP}$
        $v_i = \text{Uniform}(avg - f, Avg + f)$
endfor
for $i = 2$ to $n_{TP}$
        if $d < \text{Uniform}(0,1)$ then
                $TP_i = \text{Trend}(TP_{i-1}, V)$
        else
                $h = \arg\min_j (|TP_{i-1} - v_j|)$
                $TP_i = v_h$
                Remove $v_h$ from $V$
        endif
endfor
```

Fig. 1. Trend process

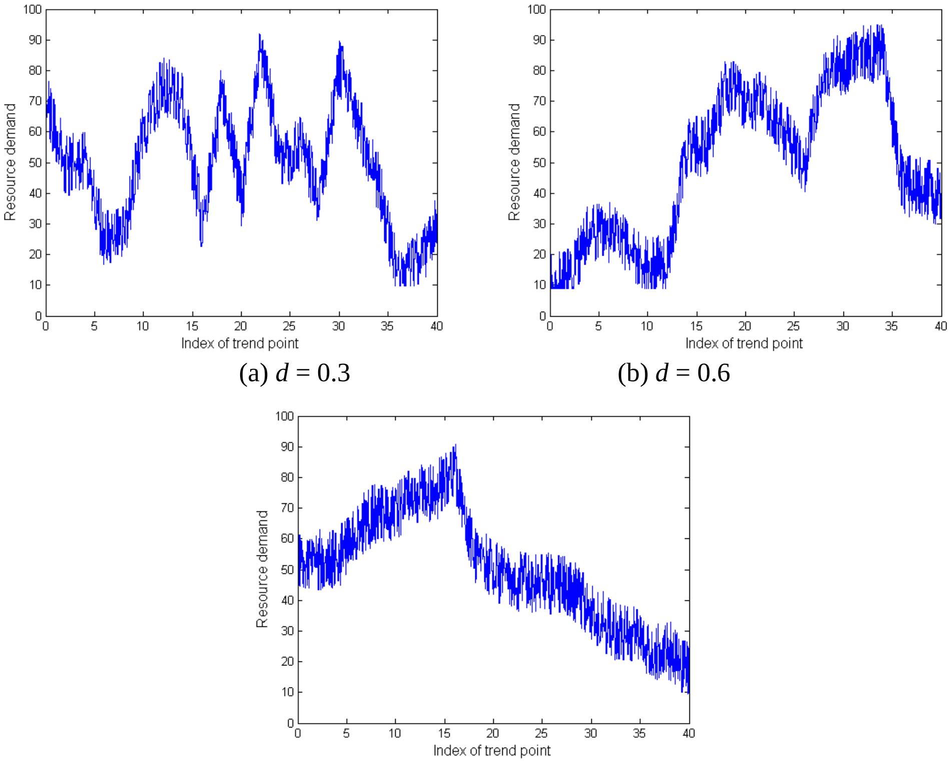
As mentioned above the residual process finalizes the resource demand patterns by adding residuals to the trends. Since a trend is in fact a line connecting trend points, the residual can be viewed as noise on the line. Therefore, we obtain residuals in a zero mean Gaussian random distribution. Let φ denote the slope of the line between two consecutive trend points and r_k denote the *k*-th residual on the same line. Then, the *k*-th resource demand value on the line, d_k, is found as follows:

$$d_k = \varphi k + r_k ,$$

A pattern of the resource demand is produced by finding the resource demands for all pairs of trend points using this formula.

We have generated resource demand patterns with different values of d in order to observe the effect of the consistency factor on the width of the variation in the resource demand. The sample patterns, which were obtained for 40 trend points with 50 residuals between every two neighboring trend points, are given in **Fig. 2**, where the resource demand values in percentage are plotted on the vertical axis, and the trend point numbers are plotted on the horizontal axis. The three patterns were generated with consistency factors of (a) 0.3, (b) 0.6, and (c) 0.9. We set the maximum resource demand to 100, the average of the resource demand values given to all trend points(avg) to 50, and the widest variation in the resource demand among all pairs of trend points(f) to 40. Note that the pattern generated with a smaller value of d reveals wider variations in the resource demand values at every pair of neighboring trend points and exhibits more peaks.

The effect of the consistency on the width of the variation in the resource demand pattern is shown in **Fig. 3**, where the average variation widths are plotted over the entire range of the values of d . We generated 100 patterns for each value of d ranging from 0 to 1 in increments of 0.1 and found the average variation in the demand values at all pairs of consecutive trend points. **Fig. 3** confirms that the variation in demand patterns decreases as the consistency increases.



(a) d = 0.3

(b) d = 0.6

(c) d = 0.9

Fig. 2. Resource demand patterns generated with different consistency values

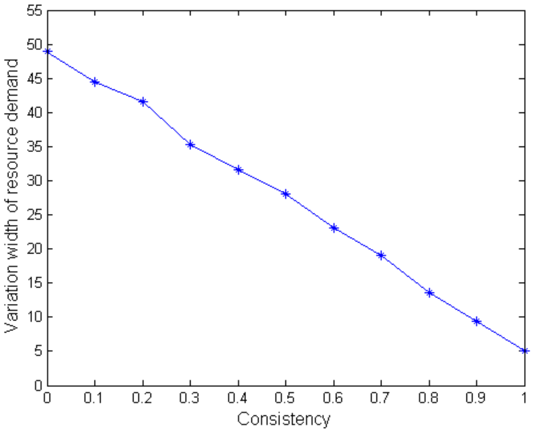


Fig. 3. Average width of variation in resource demand

4.2 Simulation

This section explains the process of the simulation that is carried out to evaluate the performance of our proposed scheme. The performance measures include the number of VM migrations taking place, the moment of migration, and the utilization of the PMs and VMs, which we obtain from the amount of resources allocated to the VMs, the resource demand pattern, and the number of VMs hosted on a PM. Our simulator operates according to the following principles:

- The resource demand of an application running on a VM is assumed to follow the generated resource demand pattern.
- If the amount of resources required by an application is greater than the resources allocated to a VM, the VM is forced to migrate to another PM that can provide the resources needed. An appropriate PM is found with the first-fit decreasing algorithm.
- Server consolidation is carried out periodically to enhance the utilization of the PMs. After each server consolidation, a PM with no VM is switched into inactive mode. When there are insufficient PMs to host the VMs, the inactive PMs are switched back into active mode for in order to host VMs.

Variables and their set values for the simulation and the demand pattern generator are given in **Table 1** and **Table 2**, respectively. The numbers of VMs and PMs were set to the same values in order to take into account the case in which every VM is allocated with the maximum amount of resources. Each demand pattern was generated with 40 trend points. Again, the average of the resource demand values given to all trend points(avg) was set to 50 units. In particular, the widest variation in the resource demand between two trend points was set to a random value between 20 units and 50 units because every application may have its own resource demand variation.

Our proposed scheme is compared with a static method and a prediction-based approach to be called the autoregression(AR)-based method in this paper. The static and the AR-based methods reserve redundant resources of 10% of the maximum amount of VM resources in order to prevent migrations that may take place due to a slight increase in the resource demand. It is assumed in our scheme that a VM can be downsized when the difference between the short- and long-term trends is less than or equal to 20 units. The width of the resource usage range is set to 10 units. The performance of the proposed scheme is analyzed with $P_A = 0.3$ and 0.7 in order to investigate the impact of P_A on the performance.

Table 1. Simulation variables

Variable	Value
No. of PMs	100
PM’s resources	100 units
No. of VMs	100
VM’s initial resources	100 units

Table 2. Variables for the resource demand pattern generator

Variable	Value
No. of trend points per pattern	40
No. of demand values between two neighboring trend points	50
Average of resource demand values(avg)	50 units
The widest variation in resource demands(f)	20~50 units

4.3 Results

This section presents the results of the analysis in comparison with the two other approaches. The graphs in this section, except for the last one, show different performance measures in relation to the varying consistency of the demand patterns. The dotted line of cross marks, the chain of stars, the solid line of circles, and the dashed line of diamonds denote the static method, the AR-based method, and the proposed scheme with acceptable migration probabilities(P_A) of 0.3 and 0.7, respectively.

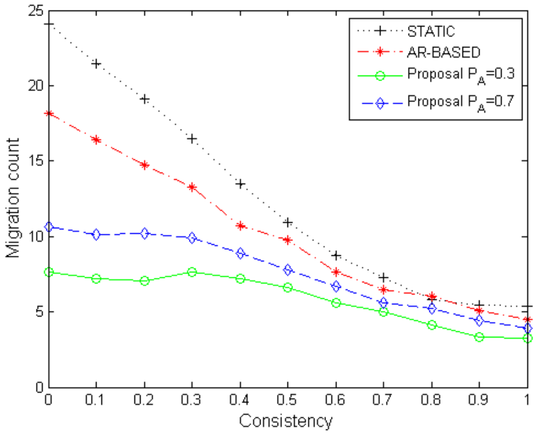


Fig. 4. Average migration count of the VMs

Fig. 4 shows the average count of VM migrations. The static method exhibits the highest frequency of migrations because it determines the reduction amount of resources based only on the resource demand of the VMs at the moment of reduction. It also shows a wide variation in migration counts depending on the consistency. The AR-based method also shows large numbers of migrations in general because it is difficult to determine the amount of reducible resources due to inaccuracy in the prediction especially when the consistency of the resource demand is low. The migration counts of our scheme are generally low and do not show much variation with different consistency values compared to the other approaches, because more extra resources remain on the VMs in order to avoid migrations. The migration count for $P_A = 0.3$ is smaller than that for $P_A = 0.7$ because a lower value of the acceptable migration probability requests conservation of more extra resources in order to keep the migration probability below the threshold. **Fig. 4** indicates that the proposed scheme features lower migration frequencies when compared to other methods.

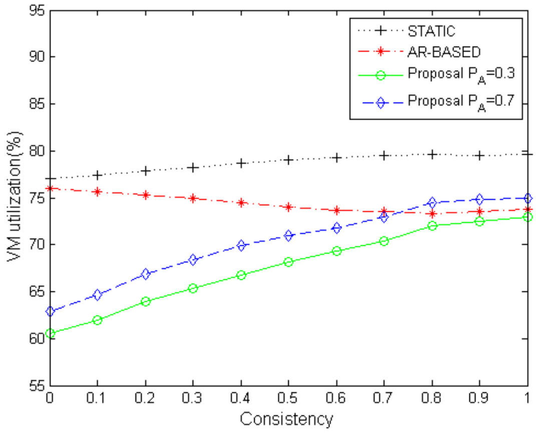


Fig. 5. Average utilization of VMs

The average utilization of VMs is given as a function of the consistency of the resource demands in **Fig. 5**. As the resource demand patterns are generated with an average of 50 units, the utilization of all of the VMs is commonly 50% if no resource reduction is made. The static method shows the highest utilization regardless of the consistency value, because it entirely reduces the excess resources of the VM, leaving only the resources needed at the time of the reduction. The AR-based method has higher utilization with lower values of consistency due to the low prediction accuracy. Its prediction accuracy decreases with lower consistency and causes improper resource reductions. Improper reductions may in turn invoke subsequent migrations, and the VM resources are reallocated, leading to higher VM utilization. Our proposed scheme shows lower VM utilization compared to other approaches because of the spare resources that are retained. However, its VM utilization increases with increased consistency because the higher consistency lowers migration probability and, hence, requires less spare resources.

Fig. 6 shows the average elapsed time after the resource reduction before the first migration occurs. The unit of the elapsed time corresponds to the interval between two consecutive residuals in the resource demand patterns. Since migration takes place when the VM resources are exhausted, a short time to the first migration after the resource reduction implies that the amount of resource reduction was calculated incorrectly. As shown in **Fig. 6**, the elapsed time of our scheme and the two other approaches commonly increases as the consistency of the demand patterns increases, starting from the shortest time with the minimum consistency. This is because miscalculation of the amount of reducible resources is more likely when there is a wider variation in the resource demands, which is simulated with a smaller value of consistency. Our scheme outperforms the other approaches in that it takes 1.2 to 2.3 times longer than the static method and 1.1 to 1.7 times longer than the AR-based method to encounter a migration after the reduction. Our scheme also shows the least variation in the elapsed time over the entire range of consistency values.

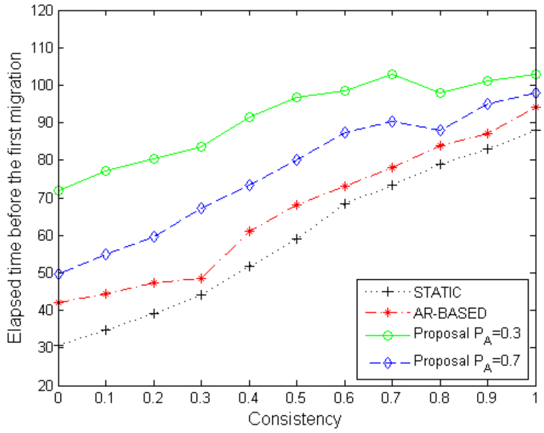


Fig. 6. Average elapsed time before the first migration

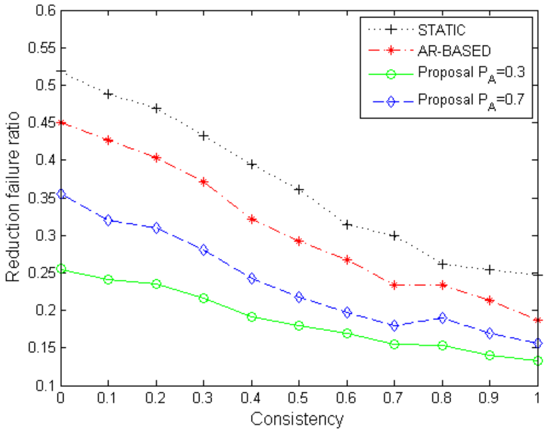


Fig. 7. Average resource reduction failure ratio

The resource reduction failure ratio is plotted in **Fig. 7**. The reduction failure ratio refers to the ratio of the number of migrations that occurred during the shortest time period when the resource usage trend can change to the total number of migrations that ever occurred. The shortest time period during which the resource usage trend can change corresponds to the interval between two neighboring trend points in a resource demand pattern. If the migration occurs during that period, then either there was no need for the resource reduction or the amount of resource reduction was miscalculated. Thus, it is defined as a reduction failure. For the case in which the consistency of the resource demand pattern is 0, the static and AR-based methods show reduction failure ratio of 51.8% and 44.9%, respectively, while our scheme shows reduction failure ratios of 35.5% and 25.5% with the acceptable migration probabilities(P_A) of 0.7 and 0.3, respectively. The overall shapes of the failure ratios of all of the schemes curves down as the consistency of the resource demand pattern increases, reaching the minimum when the consistency is set to 1. This illustrates that our scheme produces the lowest reduction failure ratio.

The result of resource reduction has an effect on the number of active PMs after server consolidation, as shown in **Fig. 8**. The static method has the smallest number of active PMs due to its highest VM utilization, as shown in **Fig. 5**. The mapping of VMs to PMs is like a bin-packing problem, where the utilization of the VMs and the number of PMs needed to host them have a negative correlation. Therefore, the number of active PMs needed to host VMs will decrease if the VMs are better utilized. As the proposed scheme has relatively low utilization compared to the other approaches due to the redundant resources needed to avoid migration, it shows a greater number of active PMs than the other approaches. With higher consistency in the resource demand patterns, however, the VM utilization of the proposed scheme increases due to the decrease in redundant resources, and the numbers of active PMs becomes smaller. When the consistency of the resource demand pattern is greater than 0.6, the difference in the number of active PMs between the proposed scheme (with $P_A = 0.7$) and the other approaches decreases. It is especially notable that the number of active PMs of our scheme (with $P_A = 0.7$) and that of the AR-based method turn in favor of our scheme as the consistency value exceeds 0.7. This is because the VM utilization of our scheme becomes higher than that of the AR-based method in the same range of consistency, as shown in **Fig. 5**.

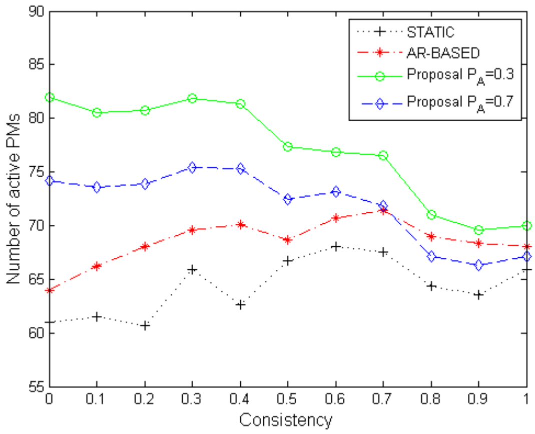


Fig. 8. Number of active PMs after server consolidation

In summary, our proposed scheme reduces the redundant resources from VMs with decreased migration frequency, but shows lower VM utilization compared to other methods, especially when the consistency of the demand pattern is low. This is because it retains a certain amount of spare resources to avoid migrations. As stated earlier, there is a tradeoff between a low migration frequency and high VM utilization, which can be balanced through the acceptable migration probability, P_A. To properly choose P_A, let us take a close look into the effect of P_A for different values of the consistency on the migration frequency and VM utilization.

Five curve lines of P_A with different values of the consistency of the resource demand are plotted in **Fig. 9** in relation to the two main performance measures of the proposed scheme, the average migration count and the average VM utilization. Please note that the smaller values of the migration count in the upper range on the vertical axis are more desirable. Ten dots on each curve indicate different values of P_A ranging from 0 to 1 in increments of 0.1. The value of 0 for P_A is indicated by the dot at the upper end of the curve and the value of 1 by the dot at the lower end.

Please note that each curve is bounded by its minimum and maximum values. This means that, with a given consistency, the migration frequency will not go below the lower limit even with P_A set to the minimum value of 0 and that the utilization will not be achieved over the upper limit even with the maximum value of 1 for a given consistency. The gaps between the two limit values of both performance measures on each curve decreases as the consistency increases.

Fig. 9 provides a guideline for the cloud service provider to choose the appropriate value of P_A when the consistency is known. For example, when the resource demand of an application shows a pattern with the consistency value around 0.4, one may choose a value of 0.6 for P_A, expecting VM utilization of 68% and a migration count of 9.3. If a value of 0.2 for P_A is chosen instead, the migration count will be improved to 6.8, while the VM utilization decreases to 65%.

Therefore, a general rule for choosing the right value of P_A is as follows. If the resource demand varies widely and irregularly in an environment in which the migration cost is high, it is desirable to set P_A to a smaller value for a lower migration frequency. On the other hand, when the resource demand shows limited variation, the VM utilization can be increased without greatly increasing the migration frequency by setting P_A to a larger value, because the difference in the migration frequency is relatively smaller than that in the utilization for a change in P_A.

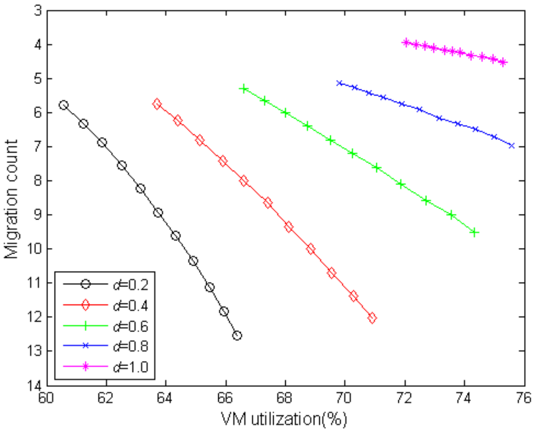


Fig. 9. Migration count and VM utilization vs. P_A and d

5. Conclusion

A VM resource reduction scheme that avoids VM migrations on a computer cluster that provides cloud service was presented. The scheme reduces the resources of a VM based on the probability that migration may occur after reduction. First, the resource usage trend of a VM is determined from the resource usage history via linear regression, and the direction of the trend and the variations in the resource usage are examined to see if the VM can be downsized. Then, the migration probability is estimated based on the VM's resource usage trend and the trend error. Finally, the amount of resources to eliminate from an underutilized VM is determined such that the migration probability after the resource reduction is less than or equal to the acceptable migration probability(P_A), which is set a priori by the service provider.

The performance of the proposed scheme was evaluated through simulation. A resource demand pattern generator has been designed and implemented to provide various resource demands for simulation. The results of the simulation showed that our reduction scheme achieves a 31.6% ~ 60.8% decrease in the migration frequency compared to other known approaches, such as the static method and the prediction-based approach, depending on the degree of consistency of the resource demands. It has also been shown that our scheme extends the elapsed time before the first migration occurrence after the resource reduction by 1.1-~ 2.3-fold, depending on the degree of consistency of the resource demands compared to those of the two other approaches. On the other hand, the VM utilization was found to decrease by 9.1~21.5% accordingly.

From the viewpoint of cloud service providers it is desirable to reduce the operational costs while maintaining the quality of service at a reasonable level. Thus, an operator-controllable parameter called the acceptable migration probability, P_A, has been introduced as a means for balancing the migration frequency and the VM utilization. For instance, P_A can be set to a lower value on resource reduction if low migration frequency takes precedence over VM utilization, and vice versa. To provide a guideline for finding an appropriate value of P_A, the effect of P_A and the consistency of the resource demands on changes in VM migration frequency and VM utilization have been investigated and presented in graphical form. It is expected that this guideline allows the service provider to enhance VM utilization while maintaining high service quality in various environments with different migration costs and operational expenses.

References

[1] H. N. Van, F. D. Tran, and J. Menaud, "Autonomy virtual resource management for service hosting platforms," in *proc. of ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pp. 1-8, 2009. Article (CrossRef Link)

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, 2010. Article (CrossRef Link)

[3] Natural Resources Defense Council "Recommendations for Tier I ENERGY STAR Computer Specification" Article (CrossRef Link)

[4] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "EnaCloud: An energy-saving application live placement approach for cloud computing environments," in *proc. of IEEE International Conference on Cloud Computing*, pp. 17-24. 2009. Article (CrossRef Link)

[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Internet Service and Applications*, vol. 1, no. 1, pp 7-18, 2010. Article (CrossRef Link)

[6] S. Lim, J. Huh, Y. Kim, and C. Das, "Migration, assignment, and scheduling of jobs in virtualized environment," Technical report, Aug. 2011. Article (CrossRef Link)

[7] M. Zhao and R. J. Figueiredo, "Experimental study of virtual machine migration in support of reservation of cluster resources," in *proc. of 2nd International Workshop on Virtualization Technology in Distributed Computing*, pp. 1–8, 2007. Article (CrossRef Link)

[8] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *proc. of 1st International Conference on Cloud Computing*, pp. 254-265, 2009. Article (CrossRef Link)

[9] C. Clark, K. Fraser, S. Hand, J. Gorm Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *proc. of 2nd Conference on Symposium on Networked Systems*

Design and Implementation, vol. 2, pp. 273-286, 2005. Article (CrossRef Link)

[10] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: A consolidation manager for cluster," in *proc. of ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pp. 41-50, 2009. Article (CrossRef Link)

[11] Y. Ho, P. Liu and J. Wu, "Server consolidation algorithms with bounded migration cost and performance guarantees in cloud computing," in *proc. of 4th IEEE International Conference on Utility and Cloud Computing*, pp. 154-161, Dec. 2011. Article (CrossRef Link)

[12] G. Jung, K. R. Joshi, M. A. Hiltunen, R. D. Schlichting, and C. Pu, "A cost-sensitive adaptation engine for server consolidation of multitier applications," in *proc. of 10th ACM/IFIP/USENIX International Conference on Middleware*, pp. 1-20, 2009. Article (CrossRef Link)

[13] A. Verma , P. Ahuja , and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *proc. of 9th ACM/IFIP/USENIX International Conference on Middleware*, pp. 243-264, 2008. Article (CrossRef Link)

[14] A. Beloglazov, J. Abawajy, and 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. Article (CrossRef Link)

[15] T. Wood, L. Cherkasova, K. Ozonat and P. Shenoy, "Profiling and modeling resource usage of virtualized applications," in *proc. of 9th ACM/IFIP/USENIX International Conference on Middleware*, pp. 366-387, 2008. Article (CrossRef Link)

[16] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper: Black-box and gray-box resource management for virtual machines," *Computer Networks*, Vol. 53, No. 17, pp. 2923-2938, 2009. Article (CrossRef Link)

[17] Z. Gong, X. Gu and J. Wilkes, "PRESS: Predictive Elastic ReSource Scaling for cloud systems," in *proc. of International Conference on Network and Server Management*, pp. 9-16, Oct. 2010. Article (CrossRef Link)

[18] A. Ganapathi, Y. Chen, A. Fox, R. Katz and D. Patterson, "Statistics-driven workload modeling for the cloud," in *proc. of IEEE International Conference on Data Engineering Workshops*, pp. 87-92, Mar. 2010. Article (CrossRef Link)

[19] W. Iqbal, M. N. Dailey, and D. Carrera, "Black-box approach to Capacity identification for multi-tier applications hosted on virtualized platforms," in *proc. of International Conference on Cloud and Service Computing*, pp. 111-117, Dec. 2011. Article (CrossRef Link)

[20] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, "Efficient resource provisioning in compute clouds via VM multiplexing," in *proc. of 7th International Conference on Autonomic Computing*, pp. 11-20, 2010. Article (CrossRef Link)

[21] T. C. Ferreto, M. Netoo, R. Calheiros, C. D. Rose, "Server consolidation with migration control for virtualized data centers," *Future Generation Computer System*, 2011. Article (CrossRef Link)

[22] S. Mehta, A. Neogi, "ReCon: A tool to recommend dynamic server consolidation in multi-cluster data centers," in *proc. of Network Operations and Management Symposium*, pp. 368-370, April 2008. Article (CrossRef Link)

[23] D. Gmach, J. Rolia, L. Cherkasova, " Resource and virtualization costs up in the cloud: Models and design choices," in *proc. of IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 395-402, June 2011. Article (CrossRef Link)

Changhyeon Kim received the B.S. degree in Computer Science and Engineering from Kyungil University, Daegu, Korea, in 2008, and the M.S. degree in Computer Science and Engineering, from Hanyang University(ERICA Campus), Ansan, Korea, in 2010. He is currently a Ph.D. candidate at Hanyang University(ERICA Campus). His research is focused on MapReduce programming model and virtualized resource management on cloud computing environment.

Wonjoo Lee received his B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University(ERICA Campus), Ansan, Korea, in 1989, 1991 and 2004, respectively. Dr. Lee joined the faculty of the Department of Computer Science at Inha Technical College, Incheon, Korea, in 2008, where he has served as the Director of the Department of Computer Science. He is currently a Professor in the Department of Computer Science, Inha Technical College. He has also served as the Vice-president of The Korean Society of Computer Information and the Editor-in-Chief for the Journal of The Korean Society of Computer Information. He is interested in parallel computing, internet and mobile computing, and cloud computing.

Changho Jeon received the B.S. degree in Electronic Engineering from Hanyang University(Seoul Campus), Seoul, Korea, in 1977, and his M.S. and Ph.D. degrees in Electrical Engineering in 1982 and 1986, respectively, from Cornell University, Ithaca, NY. Dr. Jeon has been at Hanyang University(ERICA Campus), Ansan, Korea since 1989, where he is currently a Professor in the Department of Computer Science and Engineering. He has served as the Director of the School of Electrical and Computer Engineering, the Dean of the Office of Student Supports, and the Dean of the College of Engineering Science at Hanyang University(ERICA Campus). He has also served as the Vice-president of The Korean Institute of Information Scientists and Engineers(KIISE) and the Editor-in-Chief for the KIISE magazine. His recent research field is grid/cloud computing.