Tsinghua Science and Technology

Volume 19 | Issue 1 Article 1

2014

Estimation of Cloud Node Acquisition

Waseem Ahmed
Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology (TNLIST), Tsinghua University, Beijing 100084, China, and Research Institute of Tsinghua University in Shenzhen, Shenzhen 518057, China

Yongwei Wu
Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology (TNLIST), Tsinghua University, Beijing 100084, China, and Research Institute of Tsinghua University in Shenzhen, Shenzhen 518057, China

Recommended Citation

Waseem Ahmed, Yongwei Wu. Estimation of Cloud Node Acquisition. Tsinghua Science and Technology 2014, 19(1): 1-12.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/10 pp1-12
Volume 19, Number 1, February 2014

Estimation of Cloud Node Acquisition

Waseem Ahmed and Yongwei Wu*

Abstract: Over the past decade, there has been a paradigm shift leading consumers and enterprises to the adoption of cloud computing services. Even though most cases are still in the early stages of transition, there has been a steady increase in the implementation of the pay-as-you-go or pay-as-you-grow models offered by cloud providers. Whether applied as an extension of virtual infrastructure, software, or platform as a service, many users are still challenged by the estimation of adequate resource allocation and the wide variations in pricing. Customers require a simple method of predicting future demand in terms of the number of nodes to be allocated in the cloud environment. In this paper, we review and discuss existing methodologies for estimating the demand for cloud nodes and their corresponding pricing policies. Based on our review, we propose a novel approach using the Hidden Markov Model to estimate the acquisition of cloud nodes.

Key words: cloud computing; resource allocation; hidden states; probability distribution

1 Introduction

Over the past few decade, cloud computing has gained significant popularity in providing a seamless environment to customers having large-scale distributed applications. With this emerging technology, clusters of distributed computers provide on-demand computational resources or services to end users over the Internet. As cloud computing continues to gain in popularity and usage, service providers face serious challenges in terms of scalability and complexity[1]. From the perspective of service providers, it is a big challenge to ensure the efficient usage of existing infrastructure and to accurately anticipate future demands. From the end users perspective, it is important to accurately estimate their future requirements in terms of resource acquisition by analyzing current usage patterns and performance indicators. Researchers in both academia and industry are still trying to achieve a model that can help users to accurately estimate their future resource requirements.

In general, cloud computing provides three main service delivery models, i.e., Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), and users can access these services over the Internet. Regardless of the differences in the design and technical aspects, the main difference between a Service-Oriented Architecture (SOA) and cloud computing is that the latter offers services in a pay-as-you-go manner, which means that users pay only for the services that they actually used within a certain time interval. The pricing model for these services varies depending on the agreed Quality of Service (QoS)[2]. Nonetheless, users remain skeptical as far as the performance of these services is concerned. In this paper, we discuss mainly the IaaS model because our focus is on the estimation of cloud node acquisition, where a cloud node refers to basic computing resources. Because of performance bottlenecks, users are forced to reserve sufficient resources in advance to ensure that their systems remain

• Waseem Ahmed and Yongwei Wu are with the Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology (TNLIST), Tsinghua University, Beijing 100084, China, and Research Institute of Tsinghua University in Shenzhen, Shenzhen 518057, China. E-mail: wuyw@tsinghua.edu.cn; amw.inbox@gmail.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2013-12-02; revised: 2013-12-17; accepted: 2013-12-18

up and running. However, there is no standard method for users to measure the performance of such resources, and it is difficult for users to know whether the application in a virtualized platform will subsequently perform better, or the amount of resources that will be required to ensure that the deployed application will perform as required.

IaaS refers to those services that are provided to end users in the form of processing power, storage, networking, and other basic computing resources[3]. Normally, such resources are provided to end users in the form of Virtual Machines (VMs)[4] running on physical hardware deployed in the providers data center, thus increasing resource utilization. The performance characteristics of VMs are almost similar to that of physical machines[5]. Over the last decade, virtualization technology has significantly increased in popularity. According to Huber et al.[2], the server virtualization market is expected to grow by over 30% in the coming years; nonetheless, virtualization is not simple, and users face many complex issues when managing VMs remotely. These complexities exist because (1) end users do not have access to the physical infrastructure and (2) complex interactions between workload sharing and the applications deployed on VMs have introduced new challenges for researchers. There is no standard way of gauging the status of physical machines deployed at the data centers. If a physical machine crashes, all of the VMs running on it will stop; this will in turn terminate all of the applications deployed on those VMs. Therefore, when deploying enterprise services on such VMs, there is an urgent need for an application or tool that can gauge the status of the underlying physical machines. Currently, end users can use VMs to store, deploy, and execute any application, including the OS. The number of VMs can be increased or decreased dynamically to enable a fast and easily scalable environment depending on variable workloads. The dynamic allocation of resources in a cloud environment has therefore gained significant value in the IT industry. VMs can be migrated between actual machines for better resource utilization. However, users still face performance issues during the live execution of the applications. In this paper, we discuss and highlight these issues in an IaaS environment, and then review existing methodologies that deal with such issues. We have focused specifically on the area of estimating the demand of nodes in a cloud environment to tackle performance issues during various workloads within a certain future time interval t. Then, we suggest a mechanism to predict or estimate the acquisition of cloud nodes with the help of the Hidden Markov Model (HMM) by recognizing the usage pattern of existing nodes. Our proposed method can be further utilized to select a better virtual environment during run time depending on the status of the underlying physical machines. The reason for focusing on the IaaS model is its importance in the cloud environment. The IaaS model is considered to be the foundation of all other service models, so a performance bottleneck at this level will affect all of the applications running on it. Our contribution to this paper can be summarized as follows:

- Propose a model to analyze and predict reasons for variation in VMs' performance.
- Acquisition of cloud nodes based on VMs' performance.

2 Fundamental Concepts and Terminologies

In the emerging cloud-based IT infrastructure, various providers are trying to offer innovative solutions for different kinds of enterprise applications by utilizing IaaS. The process of selecting a cloud provider(s) from a pool of options is not an easy task. For instance, Microsoft Azure, Amazon EC2, Google App Engine, and Aneka are some examples of existing cloud-based infrastructure[6]. The importance of IaaS is highlighted in a recent report Opsview 2013, where more than 16% of companies are reported to be already using IaaS and more than 30% expect to soon migrate to IaaS.

2.1 Cloud architecture

The term cloud computing refers to the parallel and distributed system environment in which users can access various interconnected virtualized computers and different kinds of applications provided by a service provider as a service. The service that is provided depends on mutually agreed terms and conditions between the cloud user and the service provider over the Internet[6]. The performance of applications and virtualized computers in real-world scenarios varies with time, and is unpredictable most of the time. Users receive a different QoS when accessing these services over the Internet. The QoS received depends on various factors, such as the quality of the network and the user load. Various studies[1,3,4,6-9] have discussed these factors in detail, while they have also ignored others, which we will discuss later in this paper. Although the cloud environment provides certain compelling features such as the absence of a capital cost, reduced operating cost, high scalability, ease of access, and lower maintenance expense[10], users still face serious problems with respect to the deployment and use of their applications in a cloud environment. Before discussing the details of the IaaS model, it is necessary to first understand the layer architecture of cloud environments. Figure 1 shows the abstract level layer architecture of parallel and distributed system environments, i.e., the cloud environment.

The first layer includes enterprise applications provided by different vendors to their clients. The clients access these applications over the Internet. Salesforce Inc. is one of the largest vendors that provide SaaS. The second layer includes a development platform for building and deploying enterprise applications. Finally, the third layer provides the virtualized hardware on the physical hardware and delivers computing power and other necessary services.

From the layered architecture, we see that actual computing resources are available on multiple data centers. Normally, each data center contains thousands of servers in the form of clusters delivering a well-defined set of services such as unlimited capacity, continuous availability, and improved efficiency[6]. These cloud resources are available to the users in the form of virtual machines that provide runtime environments for applications running on the host layer, as shown in Fig. 1. In turn, to ensure the reliability of distributed applications, we are required to examine the reliability of underlying technologies[11]. The performance of various applications running on PaaS depends on the performance of underlying computing resources. In this paper, we studied existing papers that discuss the performance of these underlying technologies, and we suggest different models to analyze the performance.

2.2 IaaS model abstract view

The basic purpose of the IaaS model is to provide users with the option of renting infrastructure resources such as computing resources, networking resources, and storage resources as a service[12]. Apart from the cost factor, the main objectives of renting these resources are to achieve high scalability and high availability. To allow the end user to manage scalability, i.e., lease resources when needed and for a specified duration, the service provider offers different interfaces that are under the end users' control. Broadly, the implementation of the IaaS model in industry can be further divided into three different categories:

- Computing as a Service;
- Storage as a Service; and
- Network as a Service.

As these resources are virtual servers or VMs on physical servers[4], as shown in Fig. 1, under Computing as a Service, service providers offer raw computing power with self-service interfaces to manage virtual servers or VMs. This helps the provider to dynamically create or terminate VMs upon request by the client. This can be done by integrating hypervisors among all VMs, as shown in Fig. 2. SaaS, which is also known as online storage, enables customers to gain online access to different storage spaces on demand. Service providers integrate all of the physical storage resources in an IaaS environment and either allocate or de-allocate virtual storage space based on Service Legal Agreement (SLA). Finally, Network as a Service allows the users to dynamically connect or disconnect virtual networks on demand in the cloud environment. This helps in dividing network request flows to different physical routers to more efficiently manage bandwidth

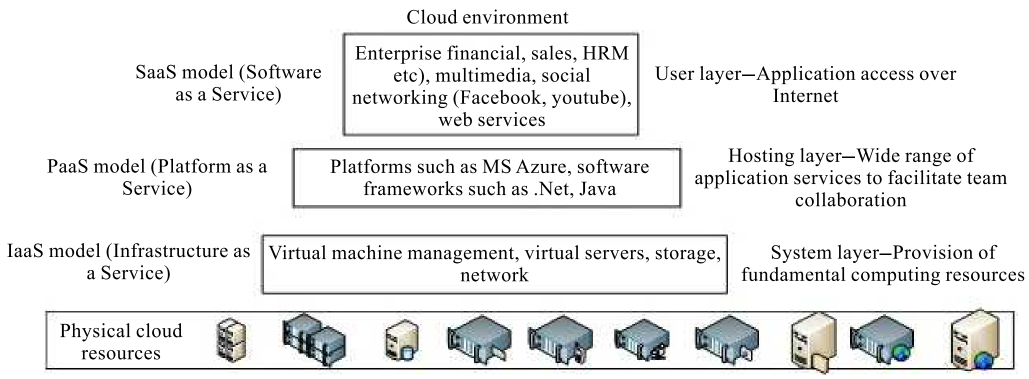

Fig. 1 Abstract level cloud computing architectural diagram.

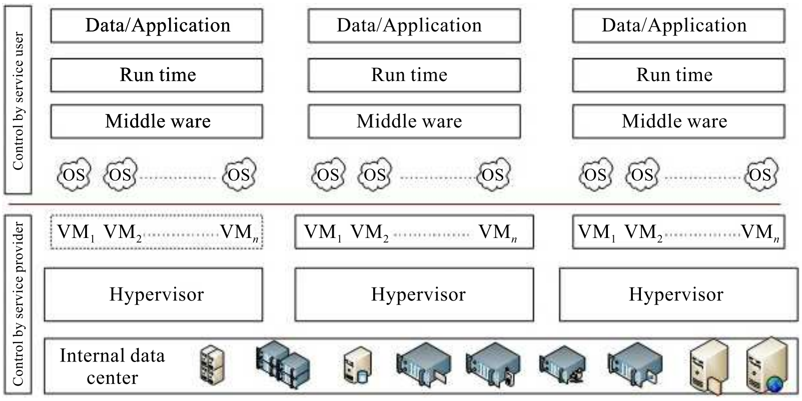

Fig. 2 IaaS layered architecture.

utilization.

The underlying goal for such a virtualization is to provide a highly available and highly reliable environment so that clients can access resources on demand without the need to consider contingency plans in the event that the server or network fails. Generally, IaaS itself has different layers that add different levels of complexity and may cause a performance bottleneck for enterprise applications running on IaaS[13]. In turn, estimating the performance of IaaS requires that the performance of underlying layers be measured, as shown in Fig. 2.

Generally, the performance of applications in cloud computing is linked with the performance of various resources such as processors, memory, storage, and networks in the cloud[8]. Figure 2 shows the layered architecture of IaaS technology, and it is obvious that even IaaS clients do not have access to the physical resources available in data centers. However, they can access different VMs running on the physical hardware[4]. Therefore, users can analyze the performance of these VMs only; however, the performance of various VMs depends on underlying physical resources present in data centers. Apart from the performance of physical resources, Armstrong and Pjemame[8] highlighted the fact that the performance of VM propagation to physical resources and the performance of para-virtualized I/O devices would also affect the overall performance of the cloud environment.

2.3 Issues with IaaS model

In cloud computing, virtualization is the most important factor that enables service providers to allocate and de-allocate enterprise level resources in a very short time at minimum cost. Although virtualization has become a new technological requirement that offers significant advantages[8], its increased usage has increased the scope of failure. To achieve the full benefits of this new technological requirement, effective planning is crucial for the successful implementation of a virtualized environment. In traditional computing systems, during a single system or hardware failure or crash, we deal with one server by transferring the load to other servers available in the cluster. However, in virtualization, the failure of a physical machine will halt all of the VMs running on it. Although it is possible to transfer the load to other available VMs, the damage incurred in this case would be comparatively larger than in the case of traditional computing systems. Apart from the failure in the physical machine, another problem in virtualization is the need to shut down all of the VMs when new hardware is to be introduced into the physical machine. Many methods are employed to overcome such issues, such as high-availability clustering[14]. However, the presence of low-performance servers and high latency may translate into the loss of customers and user frustration[15]. Therefore, this issue needs to be handled in advance to address such exceptions properly and to ensure uninterrupted access to the end users' applications.

Other areas in virtualization are also important, such as security[16, 17], VM backup strategies, and VM sprawl, which, if not addressed properly by service providers, can lead to unreliable solutions. While VMs have some good management tools, the performance optimization tools are not flexible. Therefore, the implementation of solutions on a VM is relatively easy. However, it is challenging to ensure that they have good performance. In particular, when I/O intensive applications, for instance, databases, are not addressed properly, it can lead to bottlenecks for Storage Area Networks (SANs). Similarly, as in the cloud environment, users share physical machines because multiple VMs can be deployed on one physical server; therefore, there may be sources of interference among VMs such as in the disk and network I/O[18]. This sharing can reduce the performance of VMs by increasing the run time of different tasks running on the VM.

3 Proposed Model

As discussed in Section 2.2, the IaaS model does not allow users to access actual physical machines. Instead, users have access to the VMs deployed on those machines. The failure of a physical machine can prevent access to all of the VMs running on it. As these physical machines are hidden from the users, we propose the HMM model to find the probabilistic behavior of actual physical machines. With the help of HMM, the probabilistic behavior of physical machines can be analyzed in two ways:

- **Step 1** Observe the probabilistic relation among hidden physical machines with various VMs deployed on them.
- **Step 2** Determine the probabilistic behavior of physical machines using information observed in Step 1.

3.1 Performance analysis of physical machines with HMM

HMM is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process that generates observable sequences[15]. The term hidden specifies that the internal structure of the system is hidden from the users, as shown in Fig. 1. While users do not know the state in which the system may exist, they can have a probabilistic insight of the system. In HMMs, there may exist N hidden states, a transitional probability matrix $\boldsymbol{A} = [a_{ij}]$, which stores the probabilities between hidden state j following state i, and an observation matrix $\boldsymbol{B} = [b_i(k)]$, which represents observation k being produced at any specific state i. Each state can produce any number of observable symbols based on the observation probability distribution $b_i(k)$[19] and π, which specifies the initial probability for each state. Three fundamental issues that can be resolved using HMM are evaluation, decoding, and training. More details of HMMs can be found in Ref. [15]. With HMM, the probabilistic behavior of physical machines can be observed using a two-step process as follows.

- Train the model and calculate the HMM parameters. The Baum-Welch algorithm, which is a particular case of the Expectation Maximization (EM), can be used to train the model.
- After calculating the HMM parameters, the current state of the IaaS environment can be analyzed using the VITERBI algorithm, and then based on the current state, the future state of IaaS environment is estimated using the first passage time distribution.

In our model, the underlying physical machines are referred to as hidden states, and each machine consists of N VMs that are deployed on it. The performance of each VM, such as Good (G), Normal (N), and Bad (B), is considered as an observation in our model. Users can only observe the performance of VMs in terms of G, N, and B indicators, and have no direct means of otherwise finding the performance of the underlying physical machines. When a VM performs imperfectly, this exceptional delay is believed to be because of the underlying physical machine. If we can compute the probabilistic behavior of the underlying machines, we can then predict which particular set of VMs will behave imperfectly in future during a certain time interval. Therefore, in our model we can have two assumptions:

- Performance observations (Good, Normal, Bad) of different VMs have special patterns linked with the performance of underlying machines.
- Physical machines are hidden and their number is unknown.

As HMMs have been successfully used in pattern recognition applications[15], the first assumption is based on the fact that the failure of physical machines has a direct impact on VMs that are deployed on them. Based on various failure types in physical machines, the system may therefore lead us to a specific pattern of VMs performance issues. Therefore, according to the first assumption, there is a relation between the failure types in physical machines and performance bottlenecks in VMs. By analyzing various performance bottlenecks in VMs, we can identify the failures in physical machines. The second assumption perfectly matches with the HMM, as physical machines are hidden and cannot be observed directly by the users. We can therefore define some basic parameters of the HMM in terms of the cloud environment in the following.

- States: the number of hidden states representing actual machines deployed at the data center.
- Observations: Distinct output observations, i.e., $V =$ (Normal, Delay, and Error/Crash), such that the output observation at time t is O_t, where the sequence of observations is $O{=}O_1, O_2, \cdots, O_t$. Here the sequence of observations represents the QoS attributes of VMs.
- a_{ij} represents the transitional probability from hidden state i following j.
- b_j represents the probability of a hidden state that generates outputs being produced from hidden state j.
- The initial probability distribution values of underling hidden states π.

3.2 Training the model

Generally, the HMM is trained using observation sequences, which can be formed from historical data. The purpose of training the HMM is to obtain optimal HMM parameters, which can be achieved using the Baum-Welch algorithm.

To form the training sequences, we obtained the VMs logs, and transformed the performance indicators into G, N, and B results, such that failures are represented by performance indicator B, as shown in Fig. 3. In our model, we modeled physical machines that have failures with special hidden states represented by S_{urel}, which affects the VMs, as shown in Fig. 1. Then, observation symbols G, N, and B are used as a training sequence to train the model, as shown in Table 1. The purpose of training is to adjust the HMM parameters such that the performance indicators are best represented by the model. Also, the model transits to an unreliable state whenever a performance indicator with B occurs in the training sequence.

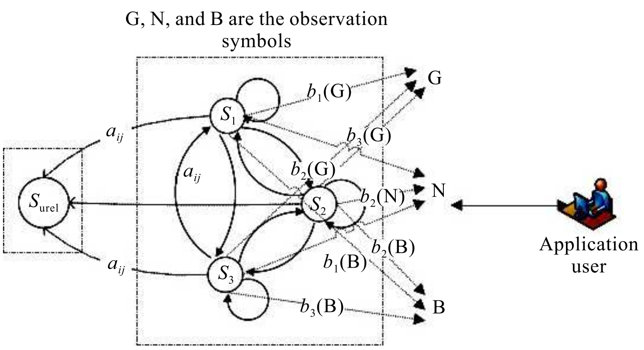

Fig. 3 Hidden states and corresponding observation symbols.

Table 1 Training sequence obtained from VMs Logs.

Sequence no.	Performance indicator	Sequence no.	Performance indicator
1	G	13	G
2	G	14	G
3	G	15	G
4	G	16	B
5	N	17	N
6	G	18	G
7	N	19	G
8	N	20	N
9	N	21	N
10	G	22	B
11	B	23	N
12	G	$\cdots$	$\cdots$

3.3 Predicting the state of the system

Predicting the state of the environment involves two steps. Firsty, compute the current state of the environment, secondly, predict the future state based on the current state. The current state of the environment can be considered as a vector of probabilities that they were in the hidden state S_i during a time interval t having performance indicators $O = O_1, O_2, \cdots, O_t$. Then, the current state of the system can lead us to predict their future state. The VITERBI algorithm in HMM helps us to compute the current state of the system, i.e.,

$$\text{Status} = \text{Max}(S_1, S_2, \cdots, S_{n-1}) \cdot P(S_1, S_2, \cdots, S_{n-1}, S_n = i, O|\lambda) \tag{1}$$

Here, "Status" represents the current status of the system, i.e., it represents the maximum probability (computed maximum of all possible hidden state sequences) that the model went through hidden states $S_1, S_2, \cdots, S_{n-1}$, and the system is in state i at hidden state n. In other words, $S_n = i$ while observing $O_1, O_2, \cdots, O_n$.

To predict the future state of the system, we can calculate $P_{\text{ur}}(n)$. Here, $P_{\text{urel}}(n)$ shows that the probability of hidden state S_k, $k = 1, 2, \cdots, n$, in the n-th time interval is unreliable, or hidden state S_k is responsible for the imperfect performance of the VM during the n-th time interval.

This probability is calculated by the First Passage Time Distribution. Let T_k be the time (known as First Passage Time in the n-th time interval during which the

VM performs imperfectly. Then, $T_k = \min(n \geqslant k \geqslant 0 : S_k = S_{\mathrm{urel}})$. Here, S_k represents the hidden state at time k during the n-th time interval. The probability distribution among the hidden states can be computed as shown below:

$$P_{\mathrm{urel}}(S_k) = \sum_{i=0}^{n} P(T_k \leqslant n | S_j = i) P(S_j = i),$$
$$\text{s.t.} \quad j = 0 \qquad (2)$$

Where $P(S_j{=}i)$ is the probability that the system is in hidden state j at the current time, as computed in Eq. (1). $P(T_k \leqslant n | S_j = i)$ is the probability of going through the hidden state during the n-th time interval starting from $j = 0$, which can be computed recursively. Equation (2) represents the probability distribution, where hidden state S_{urel} caused the VM to perform imperfectly during the time interval k, and this can be further scrutinized using dynamic programming to efficiently compute for various time intervals.

By computing the probabilistic behavior of physical machines using HMM, it is easy for the users to determine the number of nodes that will need to be allocated or de-allocated in future in the current IaaS infrastructure.

4 Existing Methodologies

The ability to estimate the acquisition of nodes at runtime in a cloud environment is an important, yet challenging research area. The performance of enterprise applications running in a virtualized environment depends mainly on three aspects, i.e., computing, storage, and networking[20]. Although the performance characteristics of VMs are almost similar to those of the underlying physical machines[5], the reliability or performance of these characteristics significantly affects the performance of applications deployed in such environments. Most studies have analyzed these attributes, and have proposed different frameworks.

4.1 Computing virtualization

Various papers[3, 10, 12, 13, 16, 21] have surveyed different issues, such as security threats and cloud node allocation policies in IaaS, and different papers have also discussed various solutions. Although cloud computing has gained enormous popularity in the solving of scalability problems, service providers still suffer from different issues regarding the scalability, such as how to ensure that different enterprise applications efficiently utilize the existing allocated resources, and how to determine when the provider should invest more resources to fulfill current demands. Buyya et al.[6] suggested a simple model to quantify the performance of resource allocation policies in cloud environments. Li et al.[4] suggested two adaptive algorithms, i.e., Adaptive List Scheduling (ALS) and Adaptive Min-Min Scheduling (AMMS) for resource allocation and task scheduling for cloud systems. In their model, they assumed a cloud system to be a heterogeneous system, and used a directed acyclic graph to schedule tasks. Caron et al.[1] and Wu et al.[22] suggested a two-stage approach using a mediator to forecast the demand for cloud computing resources. In the first stage, the mediator acquires usage probabilities from a buyer for the next period to schedule workloads, and in the second stage, the mediator predicts whether the cost can be reduced by acquiring resources over a long period. The problem with this approach is that it is based on the probabilistic values provided by the buyer. Kang et al.[23] proposed Diagnosing Application Performance Anomalies (DAPA) for virtualized infrastructures, which consists of several customized statistical techniques. The main purpose of this model is to ensure the proper allocation of resource capacities. Furthermore, this model is capable of capturing the quantitative relationship between the application performance and virtualized environments. Caron et al.[24] suggested an approach to predict the workload by identifying similar load patterns that existed previously. According to their strategy, repetitive behavioral patterns are observed in a cloud when a client accesses applications from that cloud. This behavior can be used to predict the future behavior of the same cloud. To achieve their goal, they have modified the famous Knuth-Morris-Pratt (KMP) string matching algorithm, and utilized the real-world traces from a production grid. While migrating applications from the physical infrastructure to a virtualized platform, users do not have a clear idea of whether the application will run better than before. In addition, the user does not know how many resources they need to acquire before migration to a cloud environment. Benevenuto[25] has presented a series of performance models to predict the performance of such applications that will migrate from a physical machine to a virtualized environment. They proposed a simple queuing model for the performance prediction.

Banerjee et al.[26] extended the traditional approaches for performing a reliability analysis, and proposed a model that can be used to assess the workload and reliability of SaaS applications. Their model revolves around data logs, and their study shows that data log filtration is the most important aspect when assessing and evaluating the reliability. Frncu[27] used component-based architecture to handle the workload issue in enterprise applications running in cloud environments. In their proposed method, they allocated every application component on every needed node in the cloud. According to their experiments, this will prevent the unnecessary allocation of additional nodes, and the system will continue to work even if only one node is active, giving the impression that it is a highly available system. To achieve high availability in a cloud environment, the client accesses VMs deployed on a physical server. Pearce et al.[16] and Buyya et al.[6] have discussed various issues in VM allocation policies that may lead to a performance bottleneck for different kinds of applications running in a cloud environment. Based on their defined architecture, the proposed model will utilize a better pricing model in the selection of the cloud provider. A group of researchers in academia and industry are of the view that the existing performance bottlenecks in cloud environments are mainly caused by virtualization. Barham et al.[28], Padala et al.[29], Quetier et al.[30], and Soltesz et al.[31] compared specific virtualization techniques such as full virtualization and container-based virtualization. Huber et al.[32] have proposed a systematic approach for analyzing factors influencing the performance of virtualized environments. Huber et al.[32] have proposed a generic performance prediction model for two different types of hypervisor architecture. The main purpose of their model is to estimate the performance overhead for the execution of services in a virtualized environment. Bankole and Ajila[33] have suggested a systematic model for the efficient scaling of VMs. In their model, they evaluated the TPC-W benchmark web application using three machine-learning techniques, i.e., Support Vector Machine (SVM), Neural Networks (NN), and Linear Regression (LR). Apart from the performance estimation, they identified some general factors that influence the performance, including I/O and network-intensive applications.

4.2 Storage virtualization

In a virtualized environment, I/O intensive applications such as storage applications can easily experience bottlenecks and performance degradation. Noorshams and Kounev[34] proposed a systematic approach for the analysis of the performance of I/O intensive applications. Kraft et al.[35] proposed two approaches to predict the performance of I/O intensive applications using queuing theory. In the first approach, they presented a solution for homogeneous workloads, whereas the second approach deals with heterogeneous workloads. Because resource sharing has a direct impact on the system performance, Huber et al.[36] produced a generic approach to predict the performance overhead of services running in a cloud environment. He performed his experiments on two famous hypervisor architectures, i.e., Citrix XenServer 5.5 and VMware ESX 4.0. However, the two key stakeholders, namely the system administrator and application developer, would have their own concerns as far as storage virtualization is concerned. The system administrator considers that the impact of changes in system setting is important for storage performance, while the application developer considers the response time of storage requests to be helpful in predicting the performance of the overall application. Bruhn[37] suggested a systematic performance analysis and evaluation approach for I/O intensive applications in a virtualized environment. Kundu et al.[38] proposed an artificial neural network-based model to analyze and predict the performance of virtualized environments. To study the performance of I/O intensive applications for scalable networking, Wiegert et al.[39] analyzed the performance by improving the internal setup of an I/O virtual machine monitor. In experiments, they considered various macro and micro configurations. Then, they proposed a systematic model for analyzing scalable networking for multi-core platforms. For a cloud-based real time distributed system used for online financial transactions such as banking or e-commerce, where large amounts of data are processed, there is a need for a reliable performance monitor tool. Chambliss et al.[40] proposed a Service Level Enforcement Discipline for Storage (SLEDS) controller. This controller gauges the performance of a storage system in a statistical manner. The main objective of their work was to provide a seamless environment to the users accessing the storage devices in a cloud environment. Casale et al.[41] suggested a simple model for predicting the impact of the consolidation on the storage I/O performance in the cloud. They used a measurement-based approach for the storage workload characterization, which depends on blktrace- and tshark-type tools. Their main contribution was to define simple linear prediction models for throughput, response time, and read/write requests. Kraft et al.[42] proposed a trace-driven approach for predicting the performance of storage devices in terms of the response time in a virtualized environment. They used a parameterized model, where parameters were obtained from the measurements obtained from VMs.

4.3 Network virtualization

The final aspect of the performance analysis of a cloud environment is the network infrastructure in both the virtualized and dynamic environments. Due to the large amount of load sharing and increasing density of cloud data (for instance, the large volume of network traffic over a single physical link), there is a drastic increase in the network end points in a virtualized environment[20]. This makes it difficult for the users to analyze the performance at runtime. Similarly, there is still a need for a standard and reliable model for network virtualization for cloud-based systems[20]. Existing methodologies[43,44] that evaluate the performance of network virtualization are based either on protocol-level simulation models or black-box models. These models have their own deficiencies, e.g., the black-box model does not consider the internal structure of the network, whereas the simulation model focuses only on specific parts of the network. They also ignore the link between applications and services. For instance, in Ref. [45], the authors proposed a modeling approach called Syntony. The purpose of using this approach is to model the On demand Distance Vector protocol and then to simultaneously compare the OMNeT++ implementation and model-based analysis. Similarly, to evaluate the performance of the ESRO transport protocol, de Wet and Kritzinger[46] used proSPEX. Nonetheless, in both approaches, the authors proposed a solution for modeling specific network protocols only, whereas Becker et al.[43] considered the network as a black box, and did not explore the underlying details. Following on from Ref. [43], Puigjaner[44] discussed both the classical black-box and low-level simulation approaches in his performance-modeling approach. Most of the existing methodologies have evaluated the performance at a later stage during system deployment or system execution. Nonetheless, the ability to predict the performance early in the design phase can result in savings of time, resources, and money. Huber et al.[47] proposed a Palladio Component Model based performance prediction model in an industrial environment. The model was designed to predict the performance of the system at an early stage of its development. They performed their experiments on IBM systems and validated their model by measuring data on the system z9. Verboven et al.[48] proposed a performance model for unexpected variances in the workload performance. Furthermore, to support vector machines, they suggested a novel approach using classification and regression capabilities.

4.4 Hidden Markov model

The HMM has already been used in various papers for analyzing the QoS attributes of systems running in a distributed environment. Nonetheless, they have their own issues, constraints, and shortcomings. For example, Chen et al.[49] designed a framework to evaluate the survivability of SOA-based application using the HMM. They monitored activities based on service logs or run-time statistics provided by the service provider. Nonetheless, this approach depends on the data provided by the service provider; furthermore, the author did not discuss the hidden states or probabilistic insight of WSs. Using HMM, Rahnavard et al.[50] detected anomalies in WSs. Their designed framework can be used to detect intrusions in WSs. However, this model is not applicable to gauge the QoS attribute of the overall WS. Salfner[19] categorized and distinguished patterns of errors using HMM which may lead to failures, and he also predicted the future occurrence of failures or errors. The assessment model by WS[51] used this model to assess failures during certain times in the future. Their model linked the response time variance with network states only. However, in reality, along with network states, there are other factors that can affect response time. Therefore, we considered HMM for analyzing underlying hidden states of a target web service.

In short, HMM has already been successfully used to analyze various aspects of distributed computing systems. Therefore, in this paper, we proposed a model based on HMM for estimating cloud node acquisition.

5 Conclusions

In an IaaS infrastructure, service users access VMs that are deployed on physical machines. Nonetheless, they cannot access real servers that are deployed on the companys' premises. A failure of the physical machines will terminate all of the VMs running on it, although various strategies are used to handle such issues, even though in this case, the damage would be comparatively larger than with traditional computing systems. In this paper, we have suggested a mechanism to analyze the probabilistic behavior of physical machines deployed on the service providers' premises, and then estimated the acquisition of cloud nodes by predicting their future state.

We used HMM to estimate the performance overhead of VMs by identifying the probabilistic relation among VMs and physical nodes deployed in the data center. We assumed that a cluster of physical machines contains a bad node, the failure of which will halt VMs that are running on it. Based on our prediction results, users can determine the number of VMs that can go down during a certain time period in future. Then, users can either select another cloud or acquire additional nodes in the same cloud to keep their system up and running.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61373145 and 61170210), the National High-Tech Research and Development (863) Program of China (Nos. 2013AA01A213 and 2012AA012600), and the National Key Basic Research and Development (973) Program of China (No. 2011CB302505).

References

[1] E. Caron, F. Desprez, and A. Muresan, Forecasting for grid and cloud computing on-demand resources based on pattern matching, in *Proc. IEEE CloudCom.*, 2010.

[2] N. Huber, M. von Quast, M. Hauck, and S. Kounev, Evaluating and modeling virtualization performance overhead for cloud environments, in *1st International Conference on Cloud Computing and Services Science (CLOSER 2011)*, Noordwijkerhout, The Netherlands, 2011.

[3] C. S. Pawar and R.B.Wagh, A review of resource allocation policies in cloud computing, *World Journal of Science and Technology*, vol. 2, no. 3, pp. 165-167, 2012.

[4] J. Li, M. Qiu, J. W. Niu, Y. Chen, and Z. Ming, Adaptive resource allocation for preempt able jobs in cloud systems, in *IEEE International Conference on Intelligent Systems Design and Applications*, 2010, pp. 31-36.

[5] B. Chen, C. Y. Lai, and Y. L. Huang, A performance mapping model for physical-to-virtual migration, in *Software Security and Reliability (SERE), 2013 IEEE 7th International Conference on*, 2013.

[6] R. Buyya, R. Ranjan, and R. N. Calheiros, InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services, in *Algorithms and Architectures for Parallel Processing*, Springer, 2010, pp. 13-31.

[7] Y. Demchenko, J. Ven der Ham, R. Strijkers, and M. Ghijsen, Generic architecture for cloud Infrastructure as a service (IaaS) provisioning model, Release 1. SNE Techn. Report SNE-UVA-2011-03, http://staff.science.uva.nl/∼demch/worksinprogress/sne2011-techreport-2011-03-clouds-iaasarchitecture-release1.pdf, 2011.

[8] D. Armstrong and K. Djemame, Performance issues in clouds: An evaluation of virtual image propagation and I/O paravirtualization, *The Computer Journal*, vol. 54, no. 6, pp. 836-849, 2011.

[9] A. Quiroz, H. Kim, M. Parashar, N. Gnanasambandam, and N. Sharma, Towards autonomic workload provisioning for enterprise grids and clouds, in *Grid Computing, 2009 10th IEEE/ACM International Conference on*, 2009, pp. 50-57.

[10] Q. Zhang, L. Cheng, and R. Boutaba, Cloud computing: State-of-the-art and research challenges, *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7-18, 2010.

[11] W. Ahmed and Y. Wu, A survey on reliability in distributed systems, *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1243-1255, 2013.

[12] L. Vaquero, L. Rodero-Merino, and D. Moran, Locking the sky: A survey on IaaS cloud security, *Computing*, vol. 91, no. 1, pp. 93-118, 2011.

[13] A. Iosup, R. Prodan, and D. Epema, Iaas cloud benchmarking: Approaches, challenges, and experience, in *Proc. of the Intl Conf. on High Performance Networking and Computing (SC)*, 2013.

[14] C. Procter, Virtual servers; real problems, http://www.continuitycentral.com/virtualserversrealproblems.pdf, 2007.

[15] W. Ahmed and Y. Wu, Probabilistic QoS analysis of web services, in *Network and Parallel Computing*, Springer, 2013, pp. 393-404.

[16] M. Pearce, M. Zeadally, and R. Hunt, Virtualization: Issues, security threats, and solutions, *ACM Comput. Surv.*, vol. 45, no. 2, pp. 1-39, 2013.

[17] P. Kedia, R. Nagpal, and T. P. Singh, Article: A survey on virtualization service providers, *International Journal of Computer Applications,* vol. 69, no. 24, pp. 36-42, 2013.

[18] S. Ibrahim, B. He, and H. Jin, Towards pay-as-you-consume cloud computing, in *Services Computing (SCC), 2011 IEEE International Conference on*, 2011, pp. 370-377.

[19] F. Salfner, Predicting failures with hidden Markov models, in *Proceedings of 5th European Dependable Computing Conference*, 2005.

[20] R. Piotr, Z. Steffen, and K. Samuel, A meta-model for performance modeling of dynamic virtualized network infrastructures, in *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, ACM, 2013.

[21] C. S. Yeo and R. Buyya, A taxonomy of market-based resource management systems for utility-driven cluster computing, *Software Practice and Experience*, vol. 36, no. 13, article no.13811419, 2006.

[22] F. Wu, L. Zhang, and B. Huberman, Truth-telling reservations, *Algorithmica*, vol. 52, no. 1, pp. 65-79, 2008.

[23] H. Kang, X. Zhu, and J. L. Wong, DAPA: Diagnosing application performance anomalies for virtualized infrastructures, in *Proceedings of the 2nd Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, San Jose, CA, USA, 2012, p. 16.

[24] E. Caron, F. Desprez, and A. Muresan, Forecasting for grid and cloud computing on-demand resources based on pattern matching, in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 2010, pp. 456-463.

[25] F. Benevenuto, Performance models for virtualized applications, in *Frontiers of High Performance Computing and Networking-ISPA 2006 Workshops*, Springer, 2006, pp. 427-439.

[26] S. Banerjee, H. Srikanth, and C. Bojan, Log-based reliability analysis of software as a service (SaaS), in *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on*, 2010, pp. 239-248.

[27] M. E. Frncu, Scheduling highly available applications on cloud environments, *Future Generation Computer Systems*, vol. 32, pp. 138-153, 2014.

[28] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, Xen and the art of virtualization, presented at Symposium on Operating Systems Principles, 2003.

[29] P. Padala, X. Zhu, Z. Wang, S. Singhal, and K. G. Shin, Performance evaluation of virtualization technologies for server consolidation, HP Labs Tec. Report, 2007.

[30] B. Quetier, V. Neri, and F. Cappello, Scalability comparison of four host virtualization tools, *Journal on Grid Computing*, vol. 5, no. 1, p. 8398, 2007.

[31] S. Soltesz, H. Potzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, Container-based operating system virtualization: A scalable, high-performance alternative to hypervisors, *SIGOPS Oper. Syst. Rev.,* vol. 41, no. 3, pp. 275-287, 2007.

[32] N. Huber, M. von Quast, F. Brosig, and S. Kounev, Analysis of the performance-influencing factors of virtualization platforms, *Lecture Notes in Computer Science,* vol. 6427, pp. 811-828, 2010.

[33] A. A. Bankole and S. A. Ajila, Cloud client prediction models for cloud resource provisioning in a multitier web application environment, in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium*, 2013.

[34] Q. Noorshams and S. Kounev, Experimental evaluation of the performance-influencing factors of virtualized storage systems, *Lecture Notes in Computer Science*, vol. 7587, pp. 63-79, 2013.

[35] S. Kraft, G. Casale, D. Krishnamurthy, D. Greer, and P. Kilpatrick, Performance models of storage contention in cloud environments, *Software & Systems Modeling*, vol. 12, no. 4, pp. 681-704, 2013.

[36] N. Huber, M. von Quast, F. Brosig, M. Hauck, and S. Kounev, A method for experimental analysis and modeling of virtualization performance overhead, *Cloud Computing and Services Science*, pp. 353-370, 2012.

[37] D. Bruhn, Modeling and experimental analysis of virtualized storage performance using IBM systems as example, Master's degree dissertation, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2012.

[38] S. Kundu, R. Rangaswami, K. Dutta, and M. Zhao, Application performance modeling in a virtualized environment, in *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, 2010, p. 110.

[39] J. Wiegert, G. Regnier, and J. Jackson, Challenges for scalable networking in a virtualized server, in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, 2007, pp. 179-184.

[40] D. D. Chambliss, G. A. Alvarez, P. Pandey, D. Jadav, J. Xu, R. Menon, and T. P. Lee, Performance virtualization for largescale storage systems, in *Reliable Distributed Systems, 2003. Proceedings of 22nd International Symposium on,* 2003.

[41] G. Casale, S. Kraft, and D. Krishnamurthy, A model of storage I/O performance interference in virtualized systems, in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, 2011.

[42] S. Kraft, G. Casale, D. Krishnamurthy, D. Greer, and P. Kilpatrick, IO performance prediction in consolidated virtualized environments, in *Proceedings of the 2nd ACM/SPEC International Conference on Performance Engineering*, ACM, 2011.

[43] S. Becker, H. Koziolek, and R. Reussner, The Palladio component model for model-driven performance prediction, *Journal of Systems and Software*, vol. 82, no. 1, p. 322, 2009.

[44] R. Puigjaner, Performance modelling of computer networks, in *Proc. of the IFIP/ACM Latin America Conf. on Towards a Latin American Agenda for Network Research*, New York, NY, USA, 2003, p. 106123.

[45] I. Dietrich, F. Dressler, V. Schmitt, and R. German, SYNTONY: Network protocol simulation based on standard-conform UML2 models, in *Proc. of the ValueTools 07*, 2007.

[46] N. de Wet and P. Kritzinger, Using UML models for the performance analysis of network systems, *Comput. Netw.*, vol. 49, no. 5, pp. 627-642, 2005.

[47] N. Huber, S. Becker, C. Rathfielder, J. Schweflinghaus and R. H. Reussner, Performance modeling in industry: A case study on storage virtualization, in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, Cape Town, South Africa, 2010.

[48] S. Verboven, K. Vanmechelen, and J. Broeckhove, Black box scheduling for resource intensive virtual machine workloads with interference models, *Future Generation Computer Systems,* vol. 29, no. 8, pp. 1871-1884, 2013.

[49] L. L. Chen, Q. Wang, W. Xu, and L. Zhang, Evaluating the survivability of SOA systems based on HMM, in *Proceedings of the IEEE International Conference on Web Services*, 2010.

[50] G. Rahnavard, M. S. A. Najjar, and S. Taherifar, A method to evaluate Web services anomaly detection using hidden Markov models, in *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference,* 2010.

[51] M. Zaki, A. Ihsan, and B. Athman, Web services reputation assessment using a hidden Markov model, in *Proceedings of the 7th International Joint Conference on Service-Oriented Computing*, 2009.

Waseem Ahmed received his MS degree from Quaid-e-Azam University, Islamabad, Pakistan in 2002. In the time frame of 2002 to 2010 he assumed various roles within the Software Industry in Pakistan. Since September 2010, he is following a doctoral track in Computer Science within the Parallel and Distributed Systems Group, Tsinghua University. His research interests are in the areas of performance evaluation of large-scale distributed systems, in particular service oriented architecture.

Yongwei Wu received his PhD degree in applied mathematics from the Chinese Academy of Sciences in 2002. He is currently a professor in Computer Science and Technology at Tsinghua University, China. His research interests include distributed processing, virtualization, and cloud computing. He has published over 80 research publications and has received two Best Paper Awards. He is currently on the editorial board of the International Journal of Networked and Distributed Computing and Communication of China Computer Federation. He is a member of the IEEE.