A STOCHASTIC MODEL TO INVESTIGATE DATA CENTER PERFORMANCE AND QOS IN IAAS CLOUD COMPUTING SYSTEMS

Ms. N. Thenmozhi*1, Mrs.S.Gandhimathi*2

*1Research Scholar, Dept. of CS, PGP College of Arts & Science, Namakkal, Tamilnadu, India
*2HOD/Associate Professor, Dept. of CS, PGP College of Arts & Science, Namakkal, Tamilnadu, India

Abstract—Cloud data center management is a key problem due to the numerous and heterogeneous strategies that can be applied, ranging from the VM placement to the federation with other clouds. Performance evaluation of cloud computing infrastructures is required to predict and quantify the cost-benefit of a strategy portfolio and the corresponding quality of service (QoS) experienced by users. Such analyses are not feasible by simulation or on-the-field experimentation, due to the great number of parameters that have to be investigated. In this paper, we present an analytical model, based on stochastic reward nets (SRNs), that is both scalable to model systems composed of thousands of resources and flexible to represent different policies and cloud-specific strategies. Several performance metrics are defined and evaluated to analyze the behavior of a cloud data center: utilization, availability, waiting time, and responsiveness. Our contribution includes outlining distributed middleware architecture and presenting one of its key elements: a gossip protocol that (1) ensures fair resource allocation among sites/applications, (2) dynamically adapts the allocation to load changes and (3) scales both in the number of physical machines and sites/applications. We formalize the resource allocation problem as that of dynamically maximizing the cloud utility under CPU and memory constraints. The protocol continuously executes on dynamic, local input and does not require global synchronization, as other proposed gossip protocols do. We evaluate the heuristic protocol through simulation and find its performance to be well-aligned with our design goals. A resiliency analysis is also provided to take into account load bursts. Finally, a general approach is presented that, starting from the concept of system capacity, can help system managers to opportunistically set the data center parameters under different working conditions.

Key Words: Performance Evaluation, QOS, stochastic reward net

I. INTRODUCTION

To integrate business requirements and application-level needs, in terms of quality of service (QoS), cloud service provisioning is regulated by service-level agreements (SLAs): contracts between clients and providers that express the price for a service, the QoS levels required during the service provisioning, and the penalties associated with the SLA violations. In such a context, performance evaluation plays a key role allowing system managers to evaluate the effects of different resource management strategies on the data center functioning and to predict the corresponding costs/benefits. Cloud systems differ from traditional distributed systems. First of all, they are characterized by a very large number of resources that can span different administrative domains. Moreover, the high level of resource abstraction allows us to implement particular resource management techniques such as VM multiplexing [2] or VM live migration [3] that, even if transparent to final users, have to be considered in the design of performance models to accurately understand the system behavior. Finally, different clouds, belonging to the same or to different organizations, can dynamically join each other to achieve a common goal, usually represented by the optimization of resources utilization. This mechanism, referred to as cloud federation [4], allows us to provide and release resources on demand, thus providing elastic capabilities to the whole infrastructure. For these reasons, typical performance evaluation approaches such as simulation or on-the-field measurements cannot be easily adopted. Simulation [5], [6] does not allow us to conduct comprehensive analyses of the system performance due to the great number of parameters that have to be investigated. On-the-field experiments [7], [8] are mainly focused on the offered QoS; they are based on a black box approach that makes difficult to correlate obtained data to the internal resource management strategies implemented by the system provider. On the contrary, analytical techniques [9], [10] represent a good candidate, thanks to the limited solution cost of their associated models. However, to accurately represent a cloud system, an analytical model has to be: Scalable. To deal with very large systems composed of hundreds or thousands of resources. Flexible. Allowing us to easily implement different strategies and policies and to represent different working conditions. In this paper, we
present a stochastic model, based on stochastic reward nets (SRNs) [11], that exhibits the above-mentioned features allowing to capture the key concepts of an IaaS cloud system. The proposed model is scalable enough to represent systems composed of thousands of resources and it makes possible to represent both physical and virtual resources exploiting cloud-specific concepts such as the infrastructure elasticity. With respect to the existing literature, the innovative aspect of the present work is that a generic and comprehensive view of a cloud system is presented. Low-level details, such as VM multiplexing, are easily integrated with cloud-based actions such as federation, allowing us to investigate different mixed strategies. An exhaustive set of performance metrics is defined regarding both the system provider (e.g., utilization) and the final users (e.g., responsiveness). Moreover, different working conditions are investigated and a resiliency analysis is provided to take into account the effects of load bursts. Finally, to provide a fair comparison among different resource management strategies, also taking into account the system elasticity, a performance evaluation approach is described.

II. LITERATURE SURVEY

CLOUD computing is a promising technology able to strongly modify the way computing and storage resources will be accessed in the near future. Through the provision of on-demand access to virtual resources available on the Internet, cloud systems offer services at three different levels: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). In particular, IaaS clouds provide users with computational resources in the form of virtual machine (VM) instances deployed in the provider data center, while PaaS and SaaS clouds offer services in terms of specific solution stacks and application software suites, respectively. To integrate business requirements and application-level needs, in terms of Quality of Service (QoS), cloud service provisioning is regulated by service-level agreements (SLAs): contracts between clients and providers that express the price for a service, the QoS levels required during the service provisioning, and the penalties associated with the SLA violations. In such a context, performance evaluation plays a key role allowing system managers to evaluate the effects of different resource management strategies on the data center functioning and to predict the corresponding costs/benefits.

Cloud systems differ from traditional distributed systems. First of all, they are characterized by a very large number of resources that can span different administrative domains. Moreover, the high level of resource abstraction allows implementing particular resource management techniques such as VM multiplexing or VM live migrations that, even if transparent to final users, have to be considered in the design of performance models to accurately understand the system behavior. Finally, different clouds, belonging to the same or to different organizations, can dynamically join each other to achieve a common goal, usually represented by the optimization of resources utilization. This mechanism, referred to as cloud federation, allows us to provide and release resources on demand, thus providing elastic capabilities to the whole infrastructure.

III. EXISTING SYSTEM

In order to integrate business requirements and application level needs, in terms of Quality of Service (QoS), cloud service provisioning is regulated by Service Level Agreements (SLAs): contracts between clients and providers that express the price for a service, the QoS levels required during the service provisioning, and the penalties associated with the SLA violations. In such a context, performance evaluation plays a key role allowing system managers to evaluate the effects of different resource management strategies on the data center functioning and to predict the corresponding costs/benefits.

This mechanism, referred to as cloud federation, allows providing and releasing resources on demand thus providing elastic capabilities to the whole infrastructure.

3.1 Disadvantages of existing system:

On-the-field experiments are mainly focused on the offered QoS, they are based on a black box approach that makes difficult to correlate obtained data to the internal resource management strategies implemented by the system provider.

Simulation does not allow conducting comprehensive analyses of the system performance due to the great number of parameters that have to be investigated.

IV. PROPOSED SYSTEM
We present a stochastic model, based on Stochastic Reward Nets (SRNs), that exhibits the above mentioned features allowing capturing the key concepts of an IaaS cloud system. The proposed model is scalable enough to represent systems composed of thousands of resources and it makes possible to represent both physical and virtual resources exploiting cloud specific concepts such as the infrastructure elasticity. With respect to the existing literature, the innovative aspect of the present work is that a generic and comprehensive view of a cloud system is presented. Low level details, such as VM multiplexing, are easily integrated with cloud based actions such as federation, allowing to investigate different mixed strategies.

An exhaustive set of performance metrics are defined regarding both the system provider (e.g., utilization) and the final users (e.g., responsiveness).

4.1 Advantages Of Proposed System:

To provide a fair comparison among different resource management strategies, also taking into account the system elasticity, a performance evaluation approach is described.

Such an approach, based on the concept of system capacity, presents a holistic view of a cloud system and it allows system managers to study the better solution with respect to an established goal and to opportunely set the system parameters.

V. A STOCHASTIC MODEL

SYSTEM MODELS

Among the performance models, we survey queuing systems, queuing networks, layered queuing networks, while queuing systems are widely used to model single resources subject to contention, queuing networks are able to capture the interaction among and/or application components. layered queuing networks used to better model key interaction between application mechanisms.

A. Queuing Systems

Queuing theory is commonly used in system modeling to describe hardware or software resource contention. Several analytical formulas exist, for example to characterize request mean waiting times, or waiting buffer occupancy probabilities in single queuing systems. In cloud computing, analytical queuing formulas are often integrated in optimization programs, where they are repeatedly evaluated across what-if scenarios. Common analytical formulas involve queues with exponential service and arrival times, with a single server (M/M/1) or with k servers (M/M/k), and queues with generally-distributed service times (M/G/1). Scheduling is often assumed to be firstcome first-served (FCFS) or processor sharing (PS). In particular, the M/G/1 PS queue is a common abstraction used to model a CPU and it has been adopted in many cloud studies thanks to its simplicity and the suitability to apply the model to multi-class workloads. For instance, an SLA-aware capacity allocation mechanism for cloud applications is derived using an M/G/1 PS queue as the QoS model. A resource provisioning approach of N-tier cloud web applications by modeling CPU as an M/G/1 PS queue. The M/M/1 open queue with FCFS scheduling has been used to pose constraints on the mean response time of a cloud application. Heterogeneity in customer SLAs is handled in with an M/M/k/k priority queue, which is a queue with exponentially distributed inter-arrival times and service times, k servers and no buffer. The authors use this model to investigate rejection probabilities and help dimensioning of cloud data centers. Other works that rely on queuing models to describe cloud resources include. The works in illustrate the formulation of basic queuing systems in the context of discrete-time control problems for cloud applications, where system properties such as arrival rates can change in time at discrete instants. These works show an example where a non-stationary cloud system is modeled through queuing theory. But the limitation of queuing systems is it is used to model single resources.

B. Queuing Networks

A queueing network can be described as a collection of queues interacting through request arrivals and departures. Each queue represents either a physical resource (e.g., CPU, network bandwidth, etc) or a software buffer (e.g., admission control, or connection pools). Cloud applications are often tiered and queuing networks can capture the interactions between tiers An example of cloud management solutions exploiting queuing network models is, where the cloud service center is modeled as an open queueing network of multiclass single-server queues. PS scheduling is assumed at the resources to model CPU sharing. Each layer of queues represents the collection of applications supporting the execution of requests at each tier of the cloud service center. This model is used to provide performance guarantees when defining resource allocation policies in represent a multi-tier application deployed in a cloud platform, and to derive an SLA-aware resource allocation policy. Each node in the network has exponential processing times and a generalized PS policy to approximate the operating system scheduling. The limitation with queuing systems is it is used to model the single resource.

C. Layered queuing networks

Layered queuing networks (LQNs) are an extension of queuing networks to describe layered software architectures. An LQN model of an application can be built automatically from software engineering models expressed...
using formalisms such as UML or Palladio Component Models (PCM). Compared to ordinary queuing networks, LQNs provide the ability to describe dependencies arising in a complex work-flow of requests and the layering among hardware and software resources that process them. Several evaluation techniques exist for LQNs. LQNs have been applied to cloud systems where the authors explored the impact of the network latency on the system response time for different system deployments. LQNs are here useful to handle the complexity of geo-distributed applications that include both transactional and streaming workloads. An LQN model to predict the performance of the RuBis benchmark application, which is then used as the basis of an optimization algorithm that aims at determining the best replication levels and placement of the application components. While this work is not specific to the cloud, it illustrates the application of LQNs to multitier applications that are commonly deployed in such environments. enterprise application deployed on the cloud with strict SLA requirements based on historical data. The authors also provide a discussion about the pros and cons of LQNs identifying a number of key limitations for their practical use in cloud systems. These include, among others, difficulties in modeling caching, lack of methods to compute percentiles of response times, tradeoff between accuracy and speed. Since then, evaluation techniques for LQNs that allow the computation of response time percentiles have been presented.

VI. THE ANALYTICAL MODEL

We consider an IaaS cloud system composed of N physical resources (see Fig. 1). Job requests (in terms of VM instantiation requests) are enqueued in the system queue. Such a queue has a finite size Q; once its limit is reached, further requests are rejected. The system queue is managed according to a FIFO scheduling policy. When a resource is available, a job is accepted and the corresponding VM is instantiated. We assume that the instantiation time is negligible and that the service time (i.e., the time needed to execute a job) is exponentially distributed with mean 1=\( \mu \). According to the VM multiplexing technique the cloud system can provide a number M of logical resources greater than N. In this case, multiple VMs can be allocated in the same physical machine (PM), for example, a core in a multicore architecture. Multiple VMs sharing the same PM can incur in a reduction of the performance mainly due to I/O interference between VMs. We define the degradation factor \( d(0) \) as the percentage increase in the expected service time experienced by a VM when multiplexed with another VM. The performance degradation of multiplexed VMs depends on the multiplexing technique and on the VM placement strategy. We assume that, to reduce the degradation and to obtain a fair distribution of VMs, the system is able to optimally balance the load among the PMs with respect to the resources required by VMs (e.g., trying to multiplex CPU-bound VMs only with I/O-bound VMs), thus reaching a homogeneous degradation factor. Then, indicating with \( T = \frac{1}{\mu} \) the expected service time of a VM in isolation, we can derive the expected time needed to execute two multiplexed VMs as \( T^2 \).

Modeling VM Multiplexing

When VM multiplexing is allowed, the number of running VMs can be greater than N, i.e., \( 0 \leq \# \text{run} \leq M \) and each PM can be loaded with more than one VM. Assuming an optimal scheduling algorithm able to balance the load among the N PMs, the maximum multiplexing level \( l \) reached by each PM.

The set \( J \) (with cardinality \( J \sim \frac{1}{4} \# \text{run} \)) of the instantiated VMs can be partitioned into two sets \( J_l \) and \( J_{l-1} \) (with \( J \sim \frac{1}{4} J_l \sim \frac{1}{4} J_{l-1} \)) that correspond to the set of VMs running with a multiplexing level equal to \( l \) and the set of VMs running with a multiplexing level equal to \( l-1 \), respectively.

VII. RESULT ANALYSIS

Through a transient solution of the cloud performance model of Fig. 2, it is possible to investigate the trend over time of some performance metrics. Such an analysis is straightforward to assess the resiliency of the cloud infrastructure, in particular when the load is characterized by bursts. In fact, even if the infrastructure is optimally sized with respect to the expected load, during a load burst, users can experience a degradation of the perceived QoS with corresponding violations of SLAs. For this reason, it is needed to predict the effects of a particular load condition to study the ability of the system to react to an overload situation. To study the system resiliency, we
highlight the arrival of a single burst taking into account a bursty arrival process characterized by the following behavior:

- A regular arrival rate exponentially distributed with rate $\lambda_n$, and
- A single burst with deterministic begin time $t_b$, deterministic finish time $t_f$ (and then deterministic duration given by $t_f - t_b$), and arrival rate exponentially distributed with rate $\lambda_b (> \lambda_n)$.

The bursty arrival process is modeled by opportune changing the exponentially distributed firing time of the transition $T_{arr}$ in the cloud performance model through the adoption of the technique described. First of all, we can identify three temporal phases:
1. From 0 to $t_b$: regular load.
2. From $t_b$ to $t_f$: load burst.
3. From $t_f$ to 1: regular load.

In each phase, the model is solved in transitory by setting the firing rate of $T_{arr}$ with the corresponding mean value: $\lambda_n$ for the regular load, $\lambda_b$ for the load burst. Moreover, at the beginning of each phase (i.e., before the change on the firing rate is applied), the initial state probabilities of the model have to be reloaded using the state probabilities obtained at the end of the previous phase.

**Constant Arrival Process**

We consider a cloud system characterized by a constant arrival process ($\lambda = 0.04$ job/sec). Let us suppose that a system manager is interested in increasing the system availability. We will show how the proposed model can be used to evaluate the improvements on the system availability obtained by adopting one of the above mentioned strategies, also taking into account the effects on the other performance indices. For each strategy, we increased the system capacity up to a maximum of 10 percent of the initial capacity. Fig. 5 shows the obtained results. From Fig. 5a, it can be observed that the strategy having the greatest impact on the system availability is the Federation strategy. Physical and Multiplexing strategies show similar trend even if Physical always outperforms Multiplexing, while Queuing is the solution that produces less effects. However, to effectively choose the right strategy, we have to consider other performance metrics. Fig. 5b shows the effects on the system utilization. As expected, the Federation strategy does not affect the system utilization, because it involves remote resources. On the other hand, the strategies that allow us to further increase the system utilization are Multiplexing and Queuing. Finally, it is possible to observe that an increase in physical resource (Physical strategy) will reflect on a lower steady-state utilization. From a user perspective, we will investigate the impact of the system capacity on the total service time (i.e., the sum of the waiting time and the service time) and on the system responsiveness (with a value of equal to 60 sec). From the analysis of Figs. 5c and 5d, it is possible to observe that from such a point of view the strategy that produces the most tangible effects is Physical, allowing us to both reduce the total service time and to increase the responsiveness. With respect to such metrics, Multiplexing outperforms Federation. In fact, it shows a quasiconstant trend with respect to the total service time, thanks to the balancing between the decrease on waiting time (due to the increased number of available resources) and the increase on service time (due to the increase on the multiplexing level). Moreover, the Multiplexing strategy allows us to increase the system responsiveness reaching values comparable to those of the Physical strategy. Finally, it can be observed that the Queuing strategy does not produce any benefit in terms of user satisfaction, resulting in very long waiting times and in a drastic reduction of the system responsiveness.

**VIII. CONCLUSION AND FUTURE WORK**

We have presented a stochastic model to evaluate the performance of an IaaS cloud system. Several performance metrics have been defined, such as availability, utilization, and responsiveness, allowing us to investigate the impact of different strategies on both provider and user point of views. In a market-oriented area, such as the cloud computing, an accurate evaluation of these parameters is required to quantify the offered QoS and opportunely manage SLAs. Future works will include the analysis of autonomic techniques able to change on-the-fly the system configuration to react to a change on the working conditions. We will also extend the model to represent PaaS and SaaS cloud systems and to integrate the mechanisms needed to capture VM migration and data center consolidation aspects that cover a crucial role in energy saving policies.

**REFERENCES**


