ISSN:0975-9646

A Survey on Analysis of Data centre Performance and QOS in IaaS Cloud

Karthik.M
M. Tech Student, Dept. of Computer Science
MVJ college of Engineering
Bangalore

Sharavana K
Associate Professor, Dept. of Computer Science
MVJ college of Engineering
Bangalore

***Abstract*—Recent years there is a massive migration of business applications to cloud. One of the challenges posed are data centre management and providing a survey of QOS modeling approaches and using analytical model in which stochastic reward net model, that is scalable to model systems composed of several resources like utilization, availability, waiting time is taken into consideration.**

***Keywords*—Cloud computing; quality of service; IaaS; stochastic reward nets(SRN);**

I. INTRODUCTION

The cloud system provides services at three different levels IaaS, PaaS, SaaS. IaaS provides the resources in the form of virtual machine deployed in data centre. Quality of service is very important for provisioning service level agreements. data centre performance and resource provisioning is essential. For performance analysis there are few system models has been used like queuing systems, queuing networks and layered queuing networks(LQN's) has to take into consideration. Several classes of models can be used to model QoS in cloud systems. here we briefly review queuing systems, queuing networks and layered queuing networks, however the other classes exist like stochastic reward nets and models evaluated via probabilistic model checking. The issue that a given method can perform better than other models.

II. 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 first-come 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 queueing networks can capture the interactions between tiers An example of cloud management solutions exploiting queuing network models is [55], where the cloud service center is modeled as an open queuing 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 a cloud plat- form. Also uses a queuing network to 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 multi-tier 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.

III. Proposed System

We consider an IaaS cloud system composed of N physical resources (see Fig. 1). Job requests (in terms of VM instantiation requests) are en queued 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), e.g., a core in a multi-core 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, in order 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.

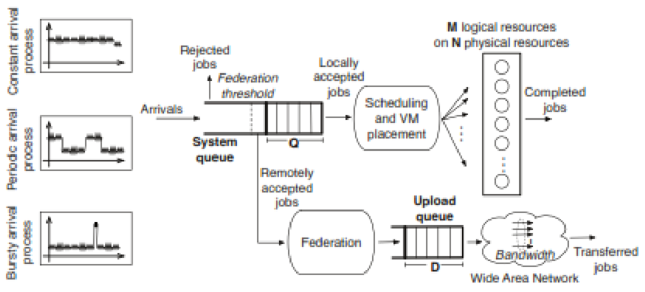


Fig. 1. An IaaS cloud system with federation.

Then, indicating with $T = 1/\mu$ the expected service time of a VM in isolation, we can derive the expected time needed to execute two multiplexed VMs as $T_2 = T \cdot (1 + d)$. In general we can express the expected execution time of i multiplexed VMs as:

$$T_i = T \cdot (1 + d)^{i-1}. \qquad (1)$$

System managers can obtain an estimation of parameter d by means of theoretical studies or statistical observations. Cloud federation allows the system to use, in particular situations, the resources offered by other public cloud systems through a sharing and paying model. In this way, elastic capabilities can be exploited in order to respond to particular load conditions. Job requests can be redirected to other clouds by transferring the corresponding VM disk images through the network. With respect to the federation technique we make the following assumptions:

- A job is redirected only if it arrives when the system queue is full.
- Federate clouds are characterized by an availability a_f.
- Federate clouds are also characterized by a quality level q_f ($0 < q_f \leq 1$) that determines the QoS reached by a request, in terms of expected service time (i.e., a VM that needs a time $T = 1/\mu$ to accomplish it works will experience an execution time $T_f = 1/(q_f \cdot \mu) \geq T$).
- A redirected job is inserted in the *upload queue* waiting for the VM transfer completion (see Fig. 1).
- There is a maximum number of concurrent redirected jobs (elasticity level) i.e., the upload queue has a finite size equal to D.
- The network bandwidth allows to transmit up to k VMs in parallel.
- The time needed to transfer a VM disk image is exponentially distributed with mean $1/\eta$.

Finally, we respect to the arrival process we will in-vestigate three different scenarios. In the first one (*Constant arrival process*) we assume the arrival process be a homogeneous Poisson process with rate λ. However, large-scale distributed systems with thousands of users, such as cloud systems, could exhibit self-similarity/long-range de-pendence with respect to the arrival process. For these reasons, in order to take into account the dependencies of the job arrival rate on both the days of a week and the hours of a day, in the second scenario (*Periodic arrival process*) we also choose to model the job arrival process as a Markov Modulated Poisson Process (MMPP). In particular, we will refer to an $MMPP$ (λ_h, λ_l, λ_{h2l}, λ_{l2h}), where λ_h and λ_l represent the expected arrival rate in high and low load conditions while $1/\lambda_{h2l}$ and $1/\lambda_{l2h}$ represent the expected duration of the two load conditions. The last scenario (*Bursty arrival process*) takes into account the presence of a burst whit fixed and short duration and it will be used in order to investigate the system resiliency.

To capture the main features of a typical IaaS cloud we make use of SRNs. SRNs are an extension of Generalized Stochastic Petri Nets (GSPNs) that allow us to associate reward rates with the marking (i.e., the distribution of tokens in the various places) . In the reminder of the paper we will use the notation $P^{\#}$ to refer to the number of token in place P . Moreover, in the function definitions we adopt a C-like syntax using the ternary operator (? :) instead of the $if - else$ construct.

The proposed SRN cloud performance model is depicted in Fig. 2.

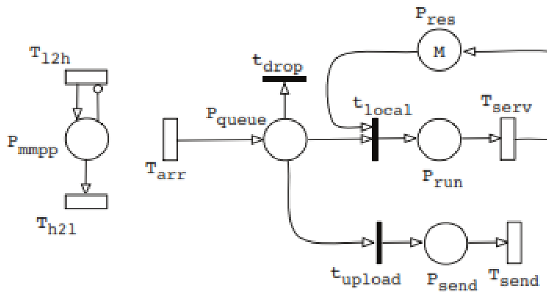


Fig. 2. The proposed SRN cloud performance model.

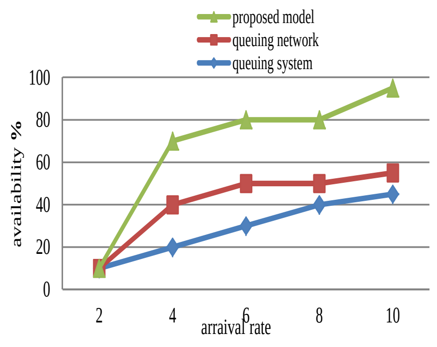
Transition T_{arr} models the arrival process. If the load (with rate equal to λ_h) when $P_{mmpp}^{\#} = 1$. The alter-nation between low and high load conditions is modeled by exponentially distributed transitions T_{h2l} and T_{l2h} with rates λ_{h2l} and λ_{l2h}, respectively. The technique adopted to model the Bursty arrival process. The system queue is modeled through place P_{queue}. A token in this place represents a job waiting in the queue. When the number of tokens in P_{queue} is greater than the queue size Q, transition t_{drop} is enabled (see the corresponding guard function in Fig. 2) thus modeling the rejection of a request.

Cloud resources are modeled by tokens in place P_{res}. Such tokens represent the M logical resources offered by the system. When a resource is available and there are pending requests, a token is removed (through transition t_{local}) from the system queue and is added to place P_{run}. If VM multiplexing is not allowed ($M = N$) service time is modeled through transition T_{serv} with firing rate equal to μ while the VM parallel execution (one for each PM) is modeled by setting transition T_{serv} with the infinite server semantic [18] in order to increase its firing rate in proportion to the number of tokens in the enabling place P_{run}. More formally, indicating with R_{serv} ($P_{run}^{\#}$) the marking dependent rate of transition T_{serv}

$$R_{serv}(P_{run}^{\#}) = P_{run}^{\#} \cdot \mu\ , 1 < P_{run}^{\#} \leq N. \qquad (2)$$

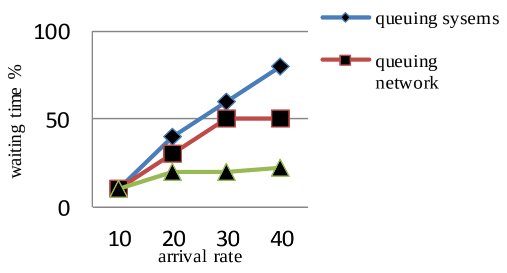
Rejection of a request. Cloud resources are modeled by tokens in place P_{res}. Such tokens represent the M logical resources offered by the system. When a resource is available and there are pending requests, a token is removed (through transition t_{local}) from the system queue and is added to place P_{run}. If VM multiplexing is not allowed ($M = N$) service time is modeled through transition T_{serv} with firing rate equal to μ while the VM parallel execution (one for each PM) is modeled by setting transition T_{serv} with the infinite server semantic in order to increase its firing rate in proportion to the number of tokens in the enabling place P_{run}. More formally, indicating with R_{serv} ($P_{run}^{\#}$) the marking dependent rate of transition T_{serv}

$$R_{serv}(P_{run}^{\#}) = P_{run}^{\#} \cdot \mu\ , 1 < P_{run}^{\#} \leq N. \qquad (3)$$



(a) Availability

The graph (a) availability shows that compare to queuing systems and the queuing networks model, the proposed stochastic reward net model will provide more availability for the cloud systems. The graph (b) waiting time shows that compare to queuing systems and queuing networks, the proposed model depicts that waiting time will be less. Here the waiting time is (delay + service time).



(b) Waiting time

IV. Conclusion

In this paper, 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 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 in order to quantify the offered QoS and op-portunely manage SLAs. Future works will include the analysis of autonomic techniques able to change on-the-fly the system configuration in order to react to a change on the working conditions. We will also extend the model in order 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

[1] R. Buyya *et al.*, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, pp. 599–616, June 2009.

[2] X. Meng *et al.*, "Efficient resource provisioning in compute clouds via vm multiplexing," in *Proceedings of the 7th international confer-ence on Autonomic computing*, ser. ICAC '10. New York, NY, USA: ACM, 2010, pp. 11–20.

[3] –131 H. Liu *et al.*, "Live virtual machine migration via asynchronous replication and state synchronization," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 12, pp. 1986 –1999, dec. 2011.

[4] B. Rochwerger *et al.*, "Reservoir - when one cloud is not enough," *Computer*, vol. 44, no. 3, pp. 44 –51, march 2011.

[5] R. Buyya, R. Ranjan, and R. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities," in *High Performance Com-puting Simulation, 2009. HPCS '09. International Conference on*, june 2009, pp. 1 –11.

[6] A. Iosup, N. Yigitbasi, and D. Epema, "On the performance variability of production cloud services," in *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on*, may 2011, pp. 104 –113.

[7] V. Stantchev, "Performance evaluation of cloud computing of-ferings," in *Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP '09. Third International Conference on*, oct. 2009, pp. 187 –192.

[8] S. Ostermann *et al.*, "A Performance Analysis of EC2 Cloud Com-puting Services for Scientific Computing," in *Cloud Computing*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2010, vol. 34, ch. 9, pp. 115.

[9] Danilo Ardagna, Giuliano Casale, Michele Ciavotta, Juan F Pérez and Weikun Wang—"Quality of service in cloud computing: modeling techniques and applications,"JisaJournal.