Performance Modeling of Cloud-based Web Systems to Estimate Response Time Distribution

Dayle Chettiar
Ryerson University
Toronto, Ontario, Canada
dayle.chettiar@ryerson.ca

Arindam Das
York University
Toronto, Ontario, Canada
raj.das.ca@gmail.com

Olivia Das
Ryerson University
Toronto, Ontario, Canada
odas@ee.ryerson.ca

ABSTRACT

Performance analysis of distributed systems with tiered software architecture has popularly entailed mean response time as the commonly used metric. It must be noted however that as a metric, response-time percentile is of greater importance since it is more desirable to reduce the variability of a system's response time, rather than minimizing the mean response time. It is a fact that analytical approximations for response time distribution do exist. However these analytical solutions capture only the steady-state behaviour (long-run behaviour) of the system. On the other hand, today's tiered cloud-based systems are so complex that they never reach steady-state. Consequently, analyzing their transient behaviour (short-term behaviour) becomes far more important than analyzing their steady-state behaviour. Regardless, it is a difficult task to accomplish transient analysis analytically due to the enormous state space of such systems. In this work, we analyze the transient behaviour of a 3-tier cloud-based system using discrete event simulation. We model the system as an open queueing network and estimate the response time distribution through the simulation. The results show that in a 3-tier system, a configuration with large number of virtual machines (VMs) does not necessarily perform better than a configuration with smaller number of VMs. The results further show that different system configurations containing the same number of VMs yield different performance depending on the replication level of software components running in different tiers. We demonstrate that our model can serve as part of a decision support system associated with dynamic VM provisioning. Our model can be used to determine whether a given number of VMs can meet the desired service-level objectives (SLOs) specified in terms of response time percentile.

CCS Concepts

• Software and its engineering→Software performance

Keywords

Performance; Discrete-event simulation; Open Queueing Network; Response time distribution.

1. INTRODUCTION

Increasingly, tiered web applications are getting deployed in clouds since cloud computing allows for dynamic scaling of computational resources as required on a pay-per-use basis. This relieves the application service providers from buying and maintaining data centers thereby reducing the operational cost. However, such deployment poses challenges for automated performance management of these applications. This is because the management system now needs to decide on the amount of computational resources (VMs) to be acquired or released dynamically for a change in system workload while ensuring that the SLOs are not violated. One way to make such a decision is to use system performance models repeatedly to evaluate various what-if scenarios [3].

System performance models can be used to predict different performance measures. The commonly used performance measure has been the mean response time (RT). However, Broadwell [5] has justified that response-time percentile as a metric is of greater importance than mean RT since it is more desirable to reduce the variability of a system's response time, rather than minimizing the mean response time.

In this work, we have used an open queueing network as our system performance model because we assume that the cloud-based systems have large number of users and the users are transient in their use of websites. Consequently, the web application might behave more like an open system as suggested by Harchol-Balter [9].

For open queueing networks, computing the exact response time distribution analytically is difficult since it may have to deal with infinite number of system states. Several approximations to response time distribution do exist though [1, 4, 6, 10, 11]. However these analytical solutions are for long-run or steady-state behaviour of the system.

Cloud-based web systems are so complex and dynamic that they never reach steady-state. Consequently, analyzing their transient behaviour becomes far more important than analyzing their steady-state behaviour [2]. In this work, we are interested in analyzing the transient behavior of the system. In spite of the importance of transient analysis, it is a difficult task to achieve it analytically due to the enormous state space of these systems. We therefore resort to discrete event simulation for our analysis.

The goal of this paper is to develop a simulation model to analyze transient behavior of a 3-tier cloud-based web system. Our model predicts the response time distribution for a given system workload. We model the system as an open queueing network with only feed-forward arcs. The system workload is represented by the arrival rate, i.e. the number of job arrivals per unit time. Here, we assume that the software server at each tier is replicated into one or more copies and each copy runs on a separate virtual machine (VM). Thus, the queueing network consists of a variable number of VMs in three tiers. Although our simulation model may be computationally expensive as compared to an analytical counterpart, it is more general in terms of service time and inter-arrival time distributions.

For our hypothetical 3-tier system, over- or under-utilization of VMs could occur if an application service provider didn't purchase enough number of VMs from the cloud provider for different tiers of the system. For example, when the number of VMs purchased is too few to handle a given workload, most of the requests will not be processed within the required response time threshold. On the contrary, if too many VMs were purchased to handle relatively fewer number of requests, the VMs will be under-utilized and this could lead to wastage of computational resources. Hence, the challenge is to find a configuration for the system consisting of the appropriate number of VMs for each stage to process the incoming requests that will ensure that a required response-time percentile is within a given threshold.

The **key contribution** of our work is twofold. *First* our work presents a model—*not only* to predict mean response time *but also* to predict the response time distribution. Our model is general enough to accommodate non-Markovian inter-arrival and service-time distributions. *Second*, our work demonstrates how our model can serve as part of a decision support system to find the appropriate configuration that would ensure that a given SLO (in terms of response-time percentile) is met.

The rest of the paper is organized as follows. Section 2 describes the 3-tier software architecture. Section 3 describes the open queueing network performance model for this architecture. Section 4 analyzes the queueing model and discusses the results. Finally section 5 concludes the work.

2. 3-TIER SOFTWARE ARCHITECTURE

Figure 1 shows the software architecture of our hypothetical 3-tier cloud-based system. We analyze this architecture in this work.

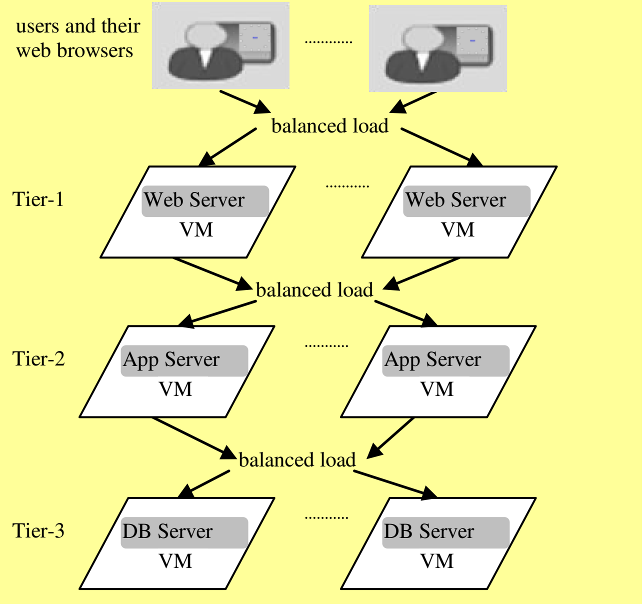

Figure 1. Software architecture of our hypothetical 3-tier cloud-based system.

The architecture consists of three tiers. One or more Web servers run in the first tier (tier-1), one or more application servers (App servers) run in the second tier (tier-2) and one or more database servers (DB servers) run in the third tier (tier-3). The users access the application at the web servers. We assume that at any given tier, one or more VMs can be provisioned, each running a single instance of a server relevant to that tier. For our modeling purposes, we assume that the workload is equally distributed among the servers at any given tier. We indicate this in Figure 1 using the phrase "balanced load".

We assume that a service request will be processed exactly once (in a server) at each tier. After completion of processing at the third tier, the response is returned to the user. We further assume that a request incurs a waiting time in the server's queue before being processed, if the server is busy. The request then incurs a service time for getting processed in the server.

The request is first sent to a Web server for processing. If the Web server is busy then the request needs to wait in the server's queue before getting processed.

The request is then redirected to an App server present in the second tier. If the App server is busy then the request needs to wait in the server's queue before being processed.

Next, the request is redirected to a DB server present in the third tier. As before, the request waits in the server's queue if the server is busy. Once the processing of the request is finished at the DB server, the response is sent back to the user.

3. SYSTEM PERFORMANCE MODEL

The 3-tier software architecture of Figure 1 is modeled as an open queueing network (see Figure 2). In Figure 2, each layer of queueing stations represents the collection of servers (each server running on its own VM) supporting execution of requests at each tier. We assume that the replicas of servers in a given tier have identical service time distribution and that the arrivals are split uniformly among them. Let λ denote the arrival rate of user requests at tier 1. If we have 3 server replicas in tier-1, then the arrival rate at each of that replica will be $\lambda/3$. We assume that μ_1 is the service rate of each Web server replica at tier-1, μ_2 denotes the service rate of each App server replica at tier-2, and μ_3 denotes the service rate of each DB Server replica at tier-3.

As shown in Figure 2, the response time of a request is the time between the arrival of the request at a tier-1 server to the completion of the request at a tier-3 server. This time includes the waiting times at the queues of the relevant servers at different tiers and the service times of those servers.

Let RT_i denote the response time of the i-th request. We assume that the SLO is specified in terms of response time percentile. An example SLO is "*The response time should be less than or equal to* 0.3 *seconds with probability* 0.95". **This means that** 95% of the requests should complete within 0.3 seconds. Here, 0.3 seconds is the response time threshold. We denote this threshold by τ.

We have simulated the open queueing network shown in Figure 2 using a discrete event simulation framework called SimPy—a Python based framework.

Let N denote the total number of requests completed in one simulation run. During every simulation run, we record the response time of each request RT_i. At the end of each simulation run, we compute the number of requests whose response time is less than or equal to the threshold τ. For request i,

$$\text{Let } n_i = 1 \text{ if } \{ RT_i \le \tau \}$$
$$= 0 \text{ otherwise}$$

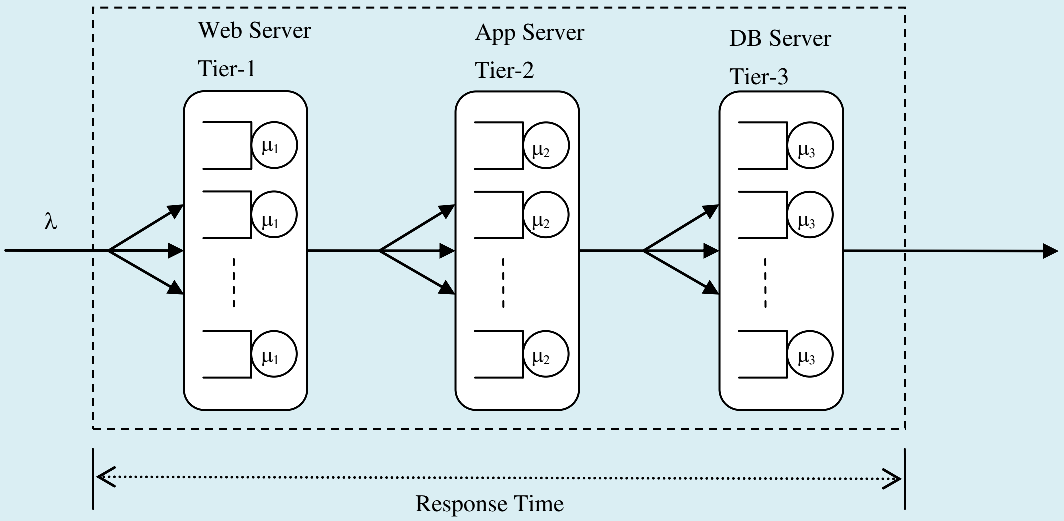

Figure 2. The 3-tier software architecture of Figure 1 depicted as an open queueing network model.

Let the random variable *RT* denote the response time. We estimate the response time distribution as:

$$P(RT \leq \tau) = \sum_{i=1}^{N} \frac{n_i}{N}$$

Further, we estimate the mean response time as:

$$Mean\ RT = \sum_{i=1}^{N} \frac{RT_i}{N}$$

4. SIMULATION RESULTS

In this section, we analyze the queueing network of Figure 2 for different configurations of VMs. We denote a VM configuration as (C_1, C_2, C_3) where C_1, C_2, C_3 denote the number of VMs in tier-1, tier-2 and tier-3 respectively. We assume that a VM in a tier runs a server replica relevant to that tier. In sub-section 4.1, we take an example VM configuration (3,2,1). Using the configuration, we estimate the response-time distribution and analyze the system's transient behavior for five different time-periods. We assume that the requests arrive according to a Poisson process and the service times of the servers are exponentially distributed. We further assume that that the system gets started with empty queues for every server. In sub-section 4.2, we demonstrate how our model can be used to evaluate various what-if scenarios in order to decide for a configuration that would meet a given SLO.

Table 1 shows the model parameters and their values for our simulation. We have adopted them from the work of Gullhav et al. [8]. We assume the arrival rate to be 100 requests/sec. We further assume that the tier-3 servers are faster than tier-2 servers, and the tier-2 servers are faster than tier-1 servers. We have assumed the service rates accordingly (see Table 1).

Table 1. Model Parameters

Parameter	Parameter Value
Arrival rate, λ	100 requests/sec
Service rate at tier-1, μ_1	60 requests/sec
Service rate at tier-2, μ_2	70 requests/sec
Service rate at tier-3, μ_3	80 requests/sec

4.1 Finding Response-time Distribution and Percentiles

To illustrate the predictions of response time distribution and percentiles, we consider the configuration (3,2,1). This configuration reflects the scenario where the database layer becomes the performance bottleneck owing to requirements of transactional access and atomicity [7]. We analyze this configuration for five different time periods: 60sec, 120sec, 180sec, 240sec and 300sec. We assume the model parameters as given in Table 1.

A plot of the five resulting response time distributions is shown in Figure 3 for the configuration (3,2,1). In this configuration, there is only one server at tier-3 which processes 80 requests per second. Since the arrival rate is 100 requests/sec, the requests get queued up in the tier-3 server as time increases. Consequently, as time passes by, more and more requests fail to meet a given threshold. If we consider an SLO specifying that "The response time should be below 15 seconds with probability 95%", then this configuration will meet the SLO for only one minute. Subsequently, it will not be able to meet the SLO any further.

Figure 4 summarizes some important statistics about the five response time distributions. It shows the mean response time and three different response time percentiles (90th, 95th and 99th) with the passage of time. We find that during a time period of 5 minutes, the configuration (3,2,1) will be able to meet a response time threshold of 57 seconds with probability 0.95.

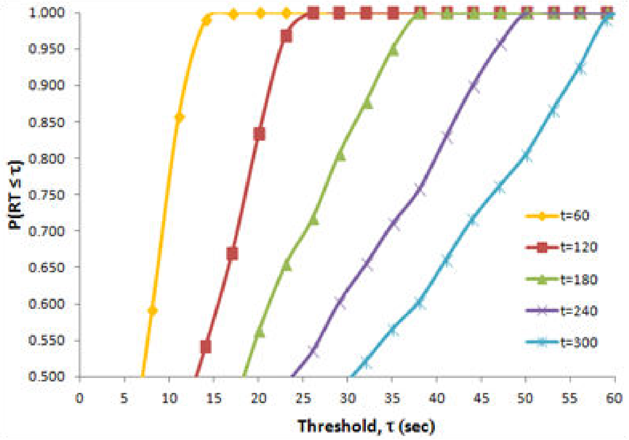

Figure 3. Transient Analysis of configuration (3,2,1): Response time distribution for five different time periods of 60, 120, 180, 240 and 300 seconds.

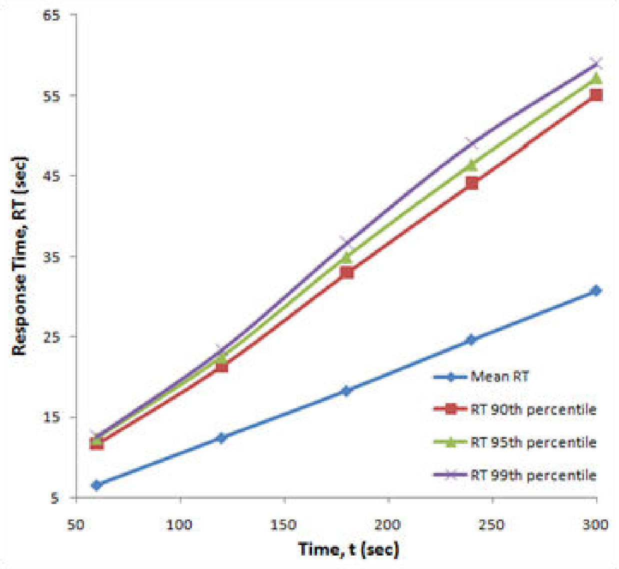

Figure 4. Mean Response Time (Mean RT), 90th, 95th and 99th percentile for RT plotted against the passage of time for the VM configuration (3,2,1). System starts with empty queues at every server.

4.2 What-if analysis in Decision Support

In this section, we illustrate how our model is used to evaluate different what-if scenarios to decide for a VM configuration that would meet a specified SLO.

We are aware that the cloud computing paradigm allows for the dynamic scaling of computational resources as required on a pay-per-use basis. Let us assume that our cost budget will allow us to buy a maximum of 3 VMs for each tier. We further assume that every VM costs the same amount of dollars. Therefore, instead of a static configuration, our aim is to change the configuration dynamically depending on the workload to be cost effective. Thus our goal is to buy a minimum number of VMs that will meet the SLO for a given workload.

Since we are allowed a maximum of 3 VMs per tier, there could be 27 potential VM configurations that could be analyzed. We analyze the response time per request for processing 1000 requests using the model parameter values provided in Table 1. We undertake this analysis for the response time threshold range of 0.1 to 0.7 seconds. We consider only those configurations for which the probability of meeting a given threshold (in the range 0.1 to 0.7 secs) is 0.55 or higher. We find that out of the 27 different configurations, only 8 of them meet this requirement.

Figure 5 shows the response time distribution for the eight configurations (2,2,2), (2,2,3), (2,3,2), (2,3,3), (3,2,2), (3,2,3), (3,3,2) and (3,3,3). Let us consider an SLO to be "The response time should be below 0.3 seconds with probability 0.95". The question we want to answer is "Which configuration is the best one to meet this SLO?" From the figure we see that the configurations (2,3,2), (2,3,3), (3,2,2), (3,2,3), (3,3,2) and (3,3,3) satisfy the SLO. However, among these configurations, (2,3,2) and (3,2,2) have smaller number of VMs (7 VMs) in comparison to others. But (2,3,2) meets the response time threshold with probability 0.969 whereas (3,2,2) meets the response time threshold with probability 0.987. So the best configuration that meets the SLO would be (3,2,2).

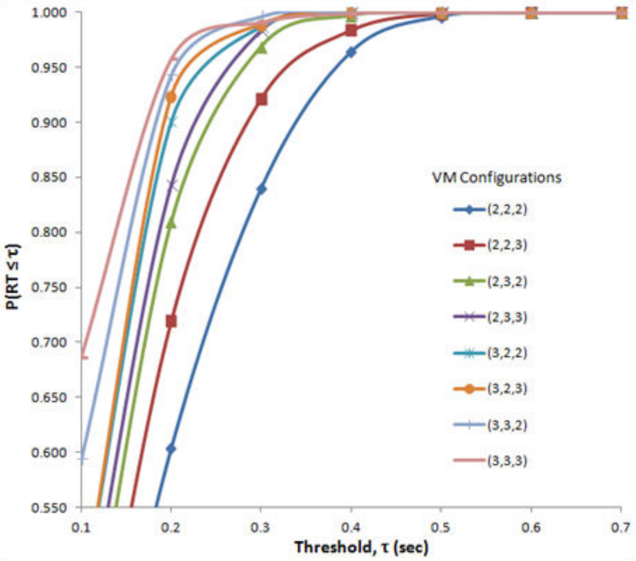

Figure 5. Response time distribution of eight different VM configurations. System starts with empty queues at every server. 1000 requests are processed.

Table 2 summarizes some important statistics about the response time distributions of eight VM configurations. It shows the mean response time and 95th and 99th response time percentiles. This table demonstrates that percentile measures are of greater importance than mean values. As an example, let us consider an SLO specified in terms of mean RT as "The mean RT should be below 0.1 seconds". Eyeballing the Table 2, we find that two configurations (3,3,2) and (3,3,3) satisfies this SLO. On the contrary, we observe in Figure 5 that the response time of 0.1

seconds will be met only by about 60% of the requests for (3,3,2), and only about 69% of the requests for (3,3,3). These percentages are way below than the 95% to 99% norm. This suggests that an SLO specified in terms of response time percentiles is more reliable than the SLO given in terms of response time averages.

Table 2. Mean Response Time (RT), 95th and 99th percentile for RT for eight configurations

VM Configuration	Mean RT (sec)	RT 95th percentile (sec)	RT 99th percentile (sec)
(2,2,2)	0.185	0.38	0.46
(2,2,3)	0.157	0.33	0.42
(2,3,2)	0.136	0.28	0.34
(2,3,3)	0.126	0.26	0.31
(3,2,2)	0.116	0.23	0.31
(3,2,3)	0.102	0.21	0.30
(3,3,2)	0.097	0.20	0.28
(3,3,3)	0.087	0.19	0.26

Next we demonstrate that different system configurations containing the same number of VMs yield different performance depending on the VM replication level in different tiers. Figure 6 shows the response time distribution for three VM configurations (2,2,3), (2,3,2) and (3,2,2). All these configurations have 7 VMs. With the assumed model parameter values of Table 1, this figure shows that the configuration (3,2,2) is better than (2,3,2) which in turn is better than (2,2,3) performance-wise. Likewise we compared sets of configurations, each set consisting of configurations having same number of VMs—sets of 4, 5, 6 and 8 VMs. We conclude that VM replication in tier-1 yields better result than replication in lower tiers among the configurations with same number of VMs.

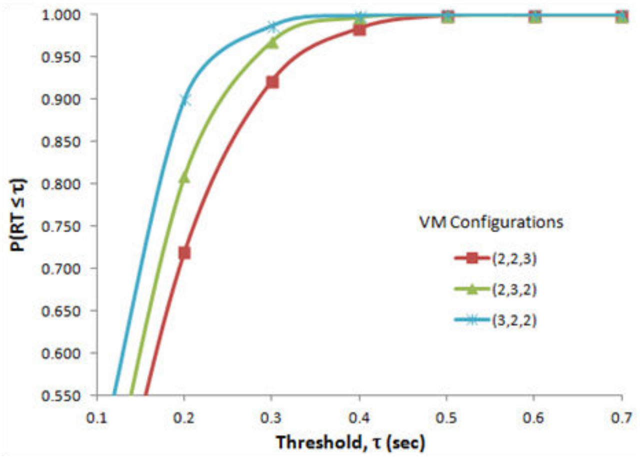

Figure 6. An example comparison of configurations with same number of VMs: Comparing three VM configurations, each having 7 VMs.

Next we demonstrate that a configuration with large number of VMs does not necessarily perform better than a configuration with smaller number of VMs. Let us consider two VM configurations: (2,3,3) with 8 VMs, and (3,2,2) with 7 VMs. With the assumed parameter values of Table 1, Figure 7 shows that the configuration (3,2,2) performs better for a response time threshold of 0.3 seconds or below than the configuration (2,3,3). Such tier-based analyses can help a service provider to refrain from unnecessarily spending money to buy excess VMs since it may not be a worthwhile investment.

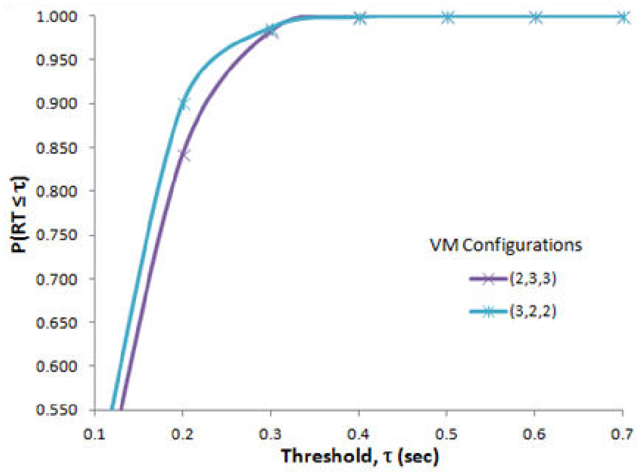

Figure 7. Large number of VMs does not necessarily lead to better performance: Comparison of VM configurations, one containing 7 VMs (2,3,3) versus the other containing 8 VMs (3,2,2).

5. CONCLUSIONS

We have developed a simulation model to analyze transient behavior of a 3-tier cloud-based web system. Our model not only predicts mean response time but also the response time percentiles. Our model is general enough to accommodate non-Markovian inter-arrival and service-time distributions. We have demonstrated how our model can serve as part of a decision support for VM planning process. Given all the VM plans satisfying SLO requirements, we acknowledge that it is not a straight forward task to figure out the optimal plan. We recommend that research be undertaken to investigate whether our model can be used jointly with an optimization engine to select the best VM plan.

6. ACKNOWLEDGMENTS

We would like to thank NSERC Canada for their financial support through NSERC Discovery Grant of Olivia Das.

7. REFERENCES

[1] Abate, J., Choudhury, G.L. and Whitt, W. 1996. Exponential approximations for tail probabilities in queues II: sojourn time and workload. Operations Research. 44, 5 (1996), 758–763.

[2] Angius, A., Horváth, A. and Wolf, V. 2013. Approximate transient analysis of queuing networks by quasi product forms. Analytical and Stochastic Modeling Techniques and Applications. Springer. 22–36.

[3] Ardagna, D., Casale, G., Ciavotta, M., Pérez, J.F. and Wang, W. 2014. Quality-of-service in cloud computing: modeling techniques and their applications. Journal of Internet Services and Applications. 5, 1 (2014), 1–17.

[4] Au-Yeung, S.W., Dingle, N.J. and Knottenbelt, W.J. 2004. Efficient approximation of response time densities and quantiles in stochastic models. ACM SIGSOFT Software Engineering Notes (2004), 151–155.

[5] Broadwell, P.M. 2004. Response time as a performability metric for online services. Report No. UCB//CSD-04-1324. Computer Science Division (EECS), University of California, Berkeley.

[6] Grottke, M., Apte, V., Trivedi, K.S. and Woolet, S. 2011. Response time distributions in networks of queues. Queueing Networks. Springer. 587–641.

[7] Grozev, N. and Buyya, R. 2014. Multi-cloud provisioning and load distribution for three-tier applications. ACM Transactions on Autonomous and Adaptive Systems (TAAS). 9, 3 (2014), 13.

[8] Gullhav, A.N., Nygreen, B. and Heegaard, P.E. 2013. Approximating the response time distribution of fault-tolerant multi-tier cloud services. Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (2013), 287–291.

[9] Harchol-Balter, M. 2013. Performance Modeling and Design of Computer Systems: Queueing Theory in Action. Cambridge University Press.

[10] Van Houdt, B. and Blondia, C. 2005. Approximated transient queue length and waiting time distributions via steady state analysis. Stochastic Models. 21, 2-3 (2005), 725–744.

[11] Van Velthoven, J., Van Houdt, B. and Blondia, C. 2005. Response time distribution in a D-MAP/PH/1 queue with general customer impatience. Stochastic Models. 21, 2-3 (2005), 745–765.