Heavy-traffic Analysis of Cloud Provisioning

Jian Tan, Hanhua Feng, Xiaoqiao Meng, Li Zhang
IBM T. J. Watson Research
Hawthorne, NY 10562, USA
{tanji,hanhfeng,xmeng,zhangli}@us.ibm.com

***Abstract*—Virtual machine (VM) provisioning is one of the fundamental components in virtualization-based cloud offerings. Modeling and analytically understanding the provisioning process is critical for the deployment and management of large-scale cloud. Based on extensive experiments on an example cloud system, we propose a queueing model to capture the important features related to scalability for the provisioning process. Specifically, we characterize how the number of VMs that can be hosted in the system and the number of physical host servers should scale according to the arriving VM requests. Note that VM provisioning incurs large I/O activities on targeted hosts with each having limited I/O resource. The logical stages during provisioning, which execute possibly on one or more physical nodes, are modeled by a semi-open Jackson Network. The model provides insights on how the performance bottlenecks can hinder the cloud scalability. Using this model we address the system sizing issue by performing heavy-traffic analysis in the classic Halfin-Whitt regime, also known as Quality and Efficiency Driven (QED), which accommodates moderate to large size cloud environments.**

I. Introduction

Cloud provisioning is the mechanism for preparing and delivering IT resources to users in cloud systems. Depending on the service types, i.e., IaaS (infrastructure as a service), PaaS (platform as a service) or SaaS (software as a service), cloud provisioning can be performed at virtual machine (VM) or application level. As IaaS has been a popular service with prominent players, e.g., Amazon EC2 and Rackspace, VM provisioning becomes an important issue that can greatly affect system performance [7], [13].

Although the implementation of VM provisioning may vary across cloud operators, the process generally consists of similar logical stages. Cloud users initiate requests for VMs with specified attributes such as the number of CPU cores, memory size and OS type. Then, the cloud management server designates a host server with sufficient capacity, and searches for a suitable VM image in the image repository. The VM image contains a byte-to-byte representation of the content of a disk and is usually stored in an external storage system. The VM image needs to be copied to the designated host on which the hypervisor creates a VM and instantiates it with the VM image. A key performance indicator for VM provisioning is the latency experienced by the user before the requested VM is ready for service. Clearly, large latencies significantly reduce user satisfaction, which in practice are often caused by non-scalable provisioning designs. Consequently, the response times of new VM requests degrade rapidly when the request volume grows beyond some critical value.

In this paper, we develop an analytical model for the VM provisioning process and use the model to provide insights on scalability. Through extensive measurements and data analysis on an IBM cloud testbed, we identify disk I/O and host capacity as two critical scalability factors. I) Because VM images are large, often above tens of GigaBytes, when multiple VM instances are provisioned on a single host, the disk I/O on the host can easily become a bottleneck; a common approach to control I/O is to reduce the number (say 1) of concurrent VM requests per physical host. II) The host capacity, measured by the total number of VMs that can be accommodated by the cloud system, also defines a hard constraint on whether a new VM request can be accepted or not. Therefore, new requests will be delayed or even dropped if disk I/O is congested or available host capacity is insufficient. Based on these observations, we propose a queueing model to study the scalability due to the limited I/O resources (embodied in the number K of active physical host servers) and host capacity (represented by the total number N of VM instances that can be hosted in the system). In our model, the logical stages during provisioning, which execute possibly on one or more physical hosts, are modeled by a semi-open Jackson Network. The model well explains how the two factors (K, N) limit the scalability. Notably, though the Markovian assumption in the Jackson model does not hold in real systems, using model parameters obtained from the measurements, we show by numerical methods that it can still serve as a good approximation even for non-Markovian measurements in computing key performance indicators.

Typical public cloud systems are designed to accommodate thousands of customers or more. Reducing operational costs, e.g., energy, is critical in maintaining a large-scale cloud system [12]. For example, when a host server does not have any VM running, it can be switched to a sleep mode or powered off to save energy. Clearly, keeping all the physical servers always active incurs huge unnecessary wastes of resources. Thus, determining the good number of physical hosts K that should remain active is crucial in balancing the efficiency and service quality. Correspondingly, it also impacts the desired host capacity N, which needs to match the workload demand.

Exact analysis of the proposed queueing model does not characterize how the system scales for moderate and large cloud environments. We therefore perform heavy-traffic analysis in the classic Halfin-Whitt regime [10], also known as Quality and Efficiency Driven (QED)regime. The Halfin-Whitt regime captures the sizing issue for typical cloud systems

using the classic square-root staffing principle that is widely used in designing call centers [6], [10], [15], [11]. It introduces safety margins for (K, N) in addition to the mean values to accommodate stochastic variability. The attractive feature of this regime is that it balances between service and economy [10], where the former emphasizes that VM requests can be accepted as early as possible (e.g., shorter waiting times) and the latter focuses on reducing the operation cost (e.g., less active host servers) of the cloud system. Intuitively, because of the random arrivals and service times, necessary safe margins of computing resources are needed for service quality protection, i.e., providing enough capacity and I/O resources. It turns out that only in the square-root scale with respect to the arrival rate can common performance metrics, e.g., VM request dropping/waiting probabilities and average waiting times, have non-trivial limits.

Usually during a one day operation cycle, typical cloud systems experience periods of high and low loads due to that customer activities often have daily patterns that change from peak hours to off-peak hours. Therefore, we can divide the whole day into a number of periods (in hours) with each period having stationary behaviors. Accordingly the heavy-traffic analysis should be interpreted as the performance in a period when the system remains stationary.

The paper is structured as follows. Section II describes the cloud provisioning process and proposes a queueing model to capture the performance bottlenecks. Then, heavy-traffic analysis is conducted in Section III to quantify the scaling law and sensitivity property. These results are illustrated in Section IV using simulation experiments.

II. Provisioning Virtual Machines

In this section, we describe our experiments on a cloud testbed used internally for generating insights for commercial product design. At a very detailed level, the provisioning process in our example system involves 245 function calls with 10 nesting levels for a single request. We then examine the more complicated situation when multiple VM requests are submitted and overlap in time. Based on the measurements, we identify the limited concurrency level, and propose a Jackson Network model to capture the important features during the provisioning process.

A. Testbed

The cloud testbed contains two X86 servers which use Kernel-based Virtual Machine (KVM) [1] as the hypervisor to create VM instances. Here the number of physical server hosts ($K = 2$) is related to the concurrency level that will be shown in Section II-B. All VM images are stored in an IBM SONAS (Scale Out Network Attached Storage) [2] system and using GPFS as the file system. The diagram of the testbed is shown in Fig. 1.

In the studied cloud testbed, VM provisioning is coordinated by a management server. The management server divides the provisioning process into hundreds of logical stages. Each stage is to accomplish a particular task, e.g., copying a VM image, creating a VM instance, assigning an IP address to the instance. A logical stage usually corresponds to a command, a shell script or a function call. Depending on the goal of the stage, a stage is handled by one or multiple processing elements in the system. These elements can be physical computers or virtual machines running on shared physical hardware. Some of the logical stages are processed with a certain concurrency level. An example is to update the resource management database which stores various resource information including IP/MAC addresses, software licenses and CPU/memory/disk usages etc. In the meanwhile, other logical stages can be processed in a distributed manner. Examples include refreshing disk partitions, resizing images and configuring virtual machines, etc.

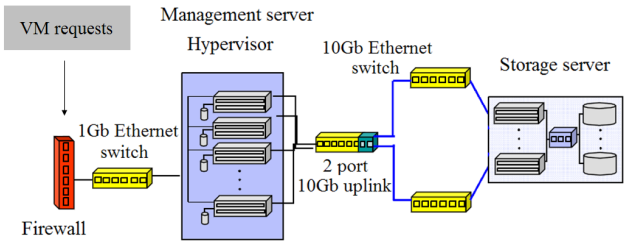


Fig. 1. VM provisioning in an example cloud system

B. Measurements

We collect timestamps at the beginning and end of each function call for every provisioning request. To visualize these nested function calls, we use a scheme that is illustrated in Fig. 2. The left bracket herein represents the starting point of a function call and the right bracket the corresponding ending point. Consecutive left or right brackets indicates the increase or decrease of the nesting level.

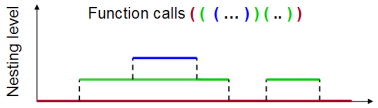


Fig. 2. Visualize nested function calls

To better understand the performance limitations we identify the concurrency level for different stages. The concurrency level at a stage is defined to be the maximum number of requests that can be in service at that stage simultaneously. This quantity is intimately related to scalability, which in many applications is difficult to obtain [3], [5], [9]. In order to avoid the performance difference caused by the variable sizes of the VM images, in our experiments, all provisioning requests target on the same image (RedHat 4.1.2-48). We plot the whole process for 9 VM requests in Fig. 3.

After a provisioning request is submitted, a few initiation steps are processed before the black vertical line, forming the first stage. The steps between the black and green vertical lines can be viewed as the next logical stage. It mainly includes updating the resource management database with information of CPU, memory, network setup, etc. for hosts

and guests and starting to copy the VM image to the host. The next logical stage between the green and blue vertical lines includes resizing of the VM image on the host. The last logical stage between the blue and red vertical lines corresponds to various configuration work. As shown in the figure, there are at most two VM requests in the second logical stage at any time. For all the other stages, many VM requests can executive concurrently at any time. The reason for the limited concurrency level of 2 for the second logical stage is because we have two KVM servers in our test bed. The provisioning software limits the concurrency level of this logical stage to be the same as the number of KVM servers.

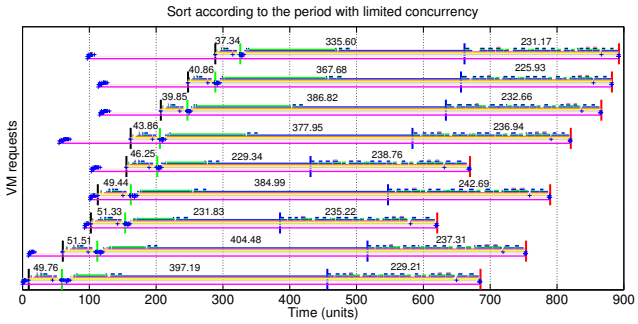


Fig. 3. The provisioning process for 9 VM requests

C. Model

From the measurements, we propose a model that captures two key features: the total number of virtual machines N that can be hosted in the system and the number of active host servers K. For N, new VM requests will be dropped when the system reaches its full capacity. For K, typical provisioning processes involves updating a resource management database and copying the targeted virtual machine image from the image pool to a virtual machine host, where the second operation usually incurs large I/O activities due to the VM image size. I/O resources are more difficult to manage than CPU and memory especially in a shared cloud environment [14]. Thus, a common approach to control I/O is to reduce the number of concurrent VM requests per physical host (say to 1). Therefore, there is a limited concurrency level due to the number of active host servers.

We propose a queueing network model, as shown in Fig. 4. The provisioning system of the cloud is modeled by two multi-

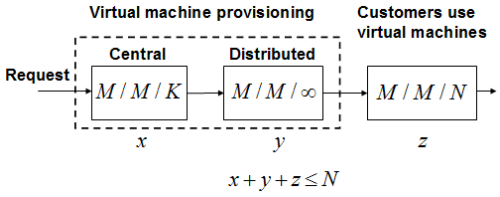


Fig. 4. VM provisioning model

server queues connected in tandem. The first $M/M/K$ queue models the limited concurrent level due to I/O constraints (embodied in the number of active host servers). For example, Fig. 3 shows that the stage between the black and green vertical lines has a concurrency level 2, which corresponds to the case $K = 2$ for the $M/M/K$ queue in our model. Recall that our testbed only has two physical host servers. In general, the number K in our model is not necessarily equal to the number of active physical hosts, but can be a function of this number. The second $M/M/\infty$ queue models the combination of other stages that can be processed in parallel in a distributed manner, e.g., configuring each individual VM instance. In this model, the VM requests arrive according to a Poisson process with constant rate λ. The first $M/M/K$ queue has a service completion rate μ and the second $M/M/\infty$ queue rate ν. The times that users spend on the virtual machines are modeled by i.i.d. exponential random variables of rate θ that are also independent of other random variables.

III. Heavy-traffic analysis of the provisioning model

Let $Q(t) = (Q_x(t), Q_y(t), Q_z(t))$ represent the number of VM requests being served at the three stations from the left to the right as shown in Fig. 4. Since the cloud system can only support at most N requests, we have $Q_x(t) + Q_y(t) + Q_z(t) \leq N$ for all $t \geq 0$. Clearly, $Q(t)$ behaves according to a continuous-time Markov chain with finite state. This model is a standard semi-open Jackson Network [8], which has a product form solution for its stationary distribution of $Q(t)$, with the state space $\{(x, y, z) : x + y + y \leq N, x, y, z \geq 0\}$. Formally we have the joint stationary distribution

$$\pi(x, y, z) = \lim_{t\to\infty} \mathbb{P}[Q_x(t) = x, Q_y(t) = y, Q_z(t) = z].$$

We can derive the normalized product form for $\pi(x, y, z)$,

$$\pi(x,y,z) = \begin{cases} \frac{\pi_0}{x!}\left(\frac{\lambda}{\mu}\right)^x \frac{1}{y!}\left(\frac{\lambda}{\nu}\right)^y \frac{1}{z!}\left(\frac{\lambda}{\theta}\right)^z, & x \leq k \\ \frac{\pi_0}{K!K^{x-K}}\left(\frac{\lambda}{\mu}\right)^x \frac{1}{y!}\left(\frac{\lambda}{\nu}\right)^y \frac{1}{z!}\left(\frac{\lambda}{\theta}\right)^z, & x > k \end{cases} \quad (1)$$

where

$$\pi_0 = \sum_{x=K+1}^{N} \sum_{0\leq y+z\leq N-x} \frac{1}{K!}\frac{1}{K^{x-K}}\left(\frac{\lambda}{\mu}\right)^x \frac{1}{y!}\left(\frac{\lambda}{\nu}\right)^y \frac{1}{z!}\left(\frac{\lambda}{\theta}\right)^z + \sum_{x\leq K, x+y+z\leq N} \frac{1}{x!}\left(\frac{\lambda}{\mu}\right)^x \frac{1}{y!}\left(\frac{\lambda}{\nu}\right)^y \frac{1}{z!}\left(\frac{\lambda}{\theta}\right)^z. \quad (2)$$

Next, we introduce three performance metrics.

1) Dropping probability: it represents the fraction of VM requests that can not be provisioned and thus have to be dropped due to lack of capacity,

$$\mathbb{P}[\text{drop}] = \sum_{x+y+z=N} \pi(x, y, z). \quad (3)$$

2) Waiting probability: it characterizes the probability that a VM request has to wait in front of the $M/M/K$ queue due to lack of available servers,

$$\mathbb{P}[\text{wait}] = \sum_{x\geq K} \pi(x, y, z). \quad (4)$$

A more precise way to define the waiting probability is through the conditional probability on the event that a request is not dropped due to full capacity by the following expression

$$\mathbb{P}[\text{wait}] = \frac{\sum_{x\geq K, x+y+z<N} \pi(x,y,z)}{\sum_{x+y+z<N} \pi(x,y,z)}. \quad (5)$$

It can be proved that (4) and (5) are asymptotically equal for large λ.

3) Average waiting time: given that a VM request can successfully start provisioning, the average waiting time W of a request before it obtains service on the $M/M/K$ queue, by Little's law, is equal to

$$\mathbb{E}[W|\text{success}] = \frac{\sum_{x\geq K} \pi(x,y,z)(x-K)}{\lambda\left(\sum_{x+y+z<N} \pi(x,y,z)\right)}. \quad (6)$$

A. Non-Markovian reality and the Jackson Network model

The service times at all queues are assumed to be exponentially distributed in our model. The measurements on the example cloud system presented in Fig. 3 show that the service times on the first and the second queue in Fig. 4 are almost equal to two constants, whereas that of the third queue (the time users spend one the virtual machines) usually exhibit heavy-tailed statistical characteristics [4]. Nevertheless, we claim that the Markovian model in Fig. 4 still serves as a very good approximation.

We support this claim by simulation experiments. Specifically, we assume constant service times of $1/\mu = 40.0$ s and $1/\nu = 600.0$ s for the first and second queues, respectively. In addition, the third queue has a service time distribution that follows a truncated Pareto distribution with a shape index of 1.5, with a minimum service time of 10.3 minutes, and a maximum of 10336.0 minutes, i.e., its probability density function is proportional to $1/x^{2.5}$ on the supporting interval with mean service time $1/\theta = 30.0$ minutes. All of the simulation experiments assume Poisson arrivals with rate λ. We plot the simulation results for $\mathbb{P}[\text{drop}]$, $\mathbb{P}[\text{wait}]$ and $\mathbb{E}[W|\text{success}]$ as functions of the arrival rate λ in Fig. 5. In this

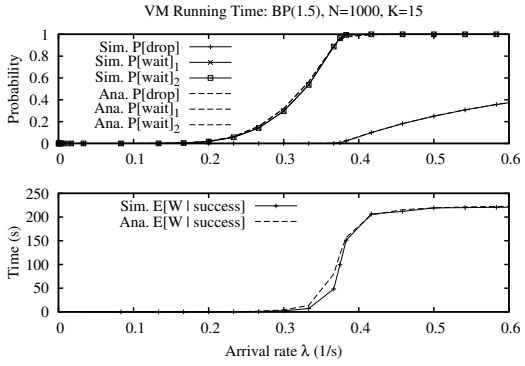


Fig. 5. Simulation and numerical results

figure, solid lines are given by the simulations and the dashed lines are analytic results computed from Equations (3), (4), (5) and (6), respectively. It is clear that the simulation results matches the analytic values very well, although we assume non-Markovian service times in the simulation. Note that the differences between the waiting times given by Eq. (4) and Eq. (5) are very small for both simulation and analytic results.

B. Asymptotic analysis in the Halfin-Whitt regime

Typical public cloud systems are designed to accommodate thousands of customers or even more. For such a large-scale system, the dimensioning problem is of vital importance for system management. However, the exact expression of the waiting and dropping probabilities in (3) and (4) can not provide the insights for the scaling law.

To this end, we follow the approach by Halfin and Whitt [10], and capture the essence of typical cloud systems in the quality and efficiency driven regime when the VM request arrival rate $\lambda \to \infty$. Specifically, we consider

$$K = \left\lfloor \frac{\lambda}{\mu} + \alpha\sqrt{\frac{\lambda}{\mu}} \right\rfloor, N = \left\lfloor \frac{\lambda}{\mu} + \frac{\lambda}{\nu} + \frac{\lambda}{\theta} + \beta\sqrt{\frac{\lambda}{\theta}} \right\rfloor, \quad (7)$$

where $-\infty < \alpha, \beta < \infty$. Recall that there are at most N virtual machines running simultaneously in the system. To simplify the analysis, we only present the cases $\alpha > 0, \beta > 0$ that are practically more interesting.

Conditions in (7) are a consequence of the classic square-root staffing principle in designing call centers [6], [10], [15], [11], which introduces the safety margins $\alpha\sqrt{\lambda/\mu}$ and $\beta\sqrt{\lambda/\theta}$ in addition to the mean values λ/μ and $\lambda/\mu + \lambda/\nu + \lambda/\theta$ for (K, N) to accommodate stochastic variability. The attractive feature of this regime is that it captures the balance between service and economy [10], where the former emphasizes that VM requests can be accepted as early as possible (e.g., small delays) and the latter is concerned about reducing the operation cost (e.g., less active host servers) of the cloud system. Roughly speaking, the average numbers of VM requests in the $M/M/K$ component and in the whole system are approximately equal to λ/μ and $\lambda/\mu+\lambda/\nu+\lambda/\theta$ according to Little's law. Because of the random arrivals and service times, the cloud system should provide a safe margin for K and N, which turn out to be in the square-root scale of λ.

Definition 1. *The Gaussian function $\varphi(x)$ is defined by $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, and the accumulated Gaussian function $\Phi(x)$ is defined by $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du$.*

Theorem 1. *Under (7), for fixed μ, ν, θ, we obtain*

$$\lim_{\lambda\to\infty} \mathbb{P}[\text{wait}] = \frac{P_3}{Q_1 + Q_2 - Q_3},$$

and

$$\lim_{\lambda\to\infty} \sqrt{\lambda}\mathbb{P}[\text{drop}] = \frac{P_1 + P_2}{Q_1 + Q_2 - Q_3},$$

$$\lim_{\lambda\to\infty} \sqrt{\lambda}\mathbb{E}[W|\text{success}] = \frac{P_4}{Q_1 + Q_2 - Q_3},$$

where

$$Q_1 = \int_{-\infty}^{\alpha} \Phi\left(\frac{\beta/\sqrt{\theta} - x/\sqrt{\nu}}{\sqrt{1/\nu + 1/\theta}}\right) d\Phi(x),$$

$$Q_2 = \frac{e^{-\alpha^2/2}}{\sqrt{2\pi}\alpha}\Phi\left(\frac{-\alpha/\sqrt{\nu}+\beta/\sqrt{\theta}}{\sqrt{1/\nu+1/\theta}}\right),$$

$$Q_3 = \frac{e^{-\alpha^2/(2\delta)}}{\sqrt{2\pi}\alpha}\Phi\left(\frac{-\alpha/(\sqrt{\nu}\delta)+\beta/\sqrt{\theta}}{\sqrt{1/\nu+1/\theta}}\right),$$

$$P_1 = \sqrt{\mu\delta}\varphi\left(\frac{\sqrt{2}\beta}{\sqrt{\theta/(\mu\delta)}}\right)\Phi\left(\frac{\alpha/\sqrt{\mu}-\beta\delta/\sqrt{\theta}}{\sqrt{\mu(1-\delta)}}\right),$$

$$P_2 = \varphi(\alpha)\sqrt{\mu}\exp\left(\alpha^2/\delta-\alpha\beta\sqrt{\frac{\mu}{\theta}}\right) \times\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}-\alpha\sqrt{\frac{\mu}{\nu}+\frac{\mu}{\theta}}\right),$$

$$P_3 = \frac{\varphi(\alpha)}{\alpha}\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)-\frac{\varphi(\alpha)}{\alpha}\exp\left(\alpha^2/\delta-\alpha\beta\sqrt{\frac{\mu}{\theta}}\right) \times\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}-\alpha\sqrt{\frac{\mu}{\nu}+\frac{\mu}{\theta}}\right).$$

$$P_4 = \frac{\varphi(\alpha)}{\sqrt{\mu}}\int_0^\infty xe^{-\alpha x}\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}-x\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)dx.$$

with $\delta \triangleq 1/(\mu/\nu+\mu/\theta+1)$.

We prove the theorem in the appendix.

Remark 1. This result shows that in the Halfin-Whitt region both $\mathbb{P}[\text{drop}]$ and $\mathbb{E}[W|\text{success}]$ are on the scale of $1/\sqrt{\lambda}$, while $\mathbb{P}[\text{wait}]$ almost does not depend on λ in a large cloud system. More importantly, these performance metrics, e.g., $\mathbb{P}[\text{wait}]$, are quite sensitive to α and β. This can also be easily seen from Fig. 5, where $\mathbb{P}[\text{wait}]$ increases very fast in the Halfin-Whitt regime. Therefore, carefully sizing the cloud system in this regime, e.g., even slightly increasing the number of host servers by $\alpha\sqrt{\lambda/\mu}$, can greatly improve the system performance. See further simulation experiments in Section IV-B.

IV. Simulation experiments

In this section we use simulation experiments to demonstrate the scaling law and the sensitivity property described in Remark 1. We use the same values for μ, ν, θ as in the experiment in Section III-A. These results can be exploited to tune α and β in (7) to satisfy certain performance requirements, which can be used to guide the management of large cloud systems.

A. Scaling law

We conduct a set of experiments to illustrate that the dropping probability and the average waiting time are on the scale of $1/\sqrt{\lambda}$ and the waiting probability is almost independent of λ in the Halfin-Whitt regime (Eq. (7)) when the system is large, as shown in Theorem 1. To see these points, we plot $\mathbb{P}[\text{drop}]\sqrt{\lambda}$, $\mathbb{E}[W|\text{success}]\sqrt{\lambda}$ and $\mathbb{P}[\text{wait}]$ in Fig. 6. As proved in Theorem 1, these three quantities are functions of α, β for fixed μ, ν, θ and independent of λ for a large cloud. Thus, for fixed α, β, we should observe a constant value for each metric. Our simulations test the 9 possible combinations for $\alpha = 0, 1, 5$ and $\beta = 0, 1, 5$, and the results indeed verify the scaling law in every case.

B. Sensitivity

Numerical calculations of the results proved in Theorem 1 show that $\mathbb{P}[\text{drop}]$, $\mathbb{E}[W|\text{success}]$ and $\mathbb{P}[\text{wait}]$ are sensitive to α or β. Therefore, sometimes even slightly increasing the number of host servers can greatly improve the system performance. This implies that careful planning and controlling the system capacity around a critical regime is very important. We demonstrate this effect in Fig. 7, where we vary α and β and fix all other parameters. For example, the waiting probability shown in Fig. 7(c) decreases from 0.21 to 0.01 even when α only increases from -1.0 to 1.0.

V. Conclusion

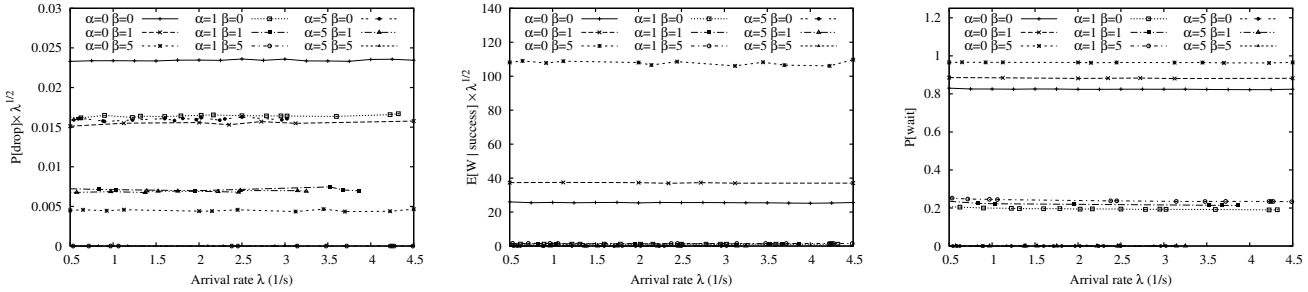
Based on a real example cloud system, we carefully examine the VM provisioning process. Using the collected measurements we identify the period with limited concurrency level. From these measurements we propose a queueing model based on Jackson Network to capture two important features, i.e., the limited I/O resources and user capacity, which determine the scalability of the cloud system. Simulation experiments show that even in real settings when Markovian assumptions do not hold, our proposed Jackson Network model still serves as a good approximation. Since the exact solution can not provide insights for the scaling law when the system grows, we conduct the heavy-traffic analysis in the classic Halfin-Whitt regime, which accommodates moderate to large cloud environments. We demonstrate that the VM request dropping probability and the average waiting time is on the scale of square root of the number of physical hosts, while the waiting probability is quite sensitive to the number of host servers. These analytical results show that carefully sizing the cloud system in this regime, e.g., even slightly increasing the number of host servers around critical points, can greatly improve the system performance.

Appendix

Proof of Theorem 1: In order to ease the presentation of the proof, we first introduce some definitions. Define $\rho = \lambda/(K\mu)$ and the following expressions:

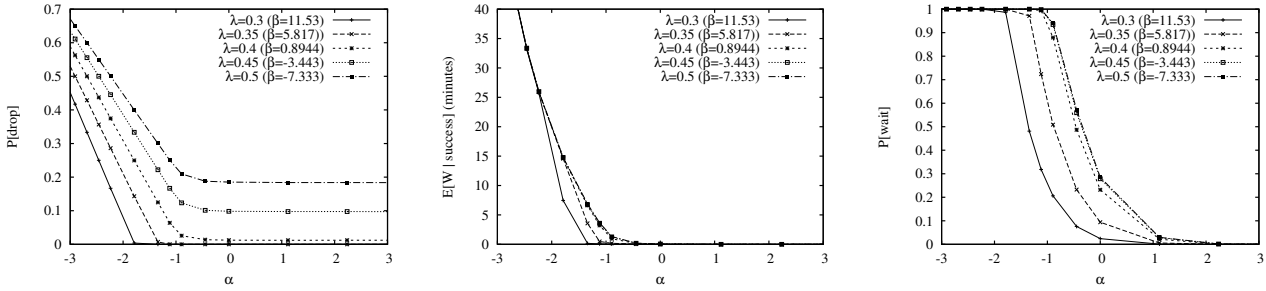
$$Q_1^\lambda \triangleq \left(\sum_{x\le K, x+y+z\le N}\frac{1}{x!}\left(\frac{\lambda}{\mu}\right)^x\frac{1}{y!}\left(\frac{\lambda}{\nu}\right)^y\frac{1}{z!}\left(\frac{\lambda}{\theta}\right)^z\right) \times\exp\left(-\left(\frac{\lambda}{\mu}+\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)\right), \quad (8)$$

$$Q_2^\lambda \triangleq \left(\sum_{j=0}^{N-K-1}\frac{1}{j!}\left(\frac{\lambda}{\mu}+\frac{\lambda}{\theta}\right)^j\frac{1}{K!}\frac{1}{K}\left(\frac{\lambda}{\mu}\right)^{K+1}\frac{1}{1-\rho}\right) \times\exp\left(-\left(\frac{\lambda}{\mu}+\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)\right), \quad (9)$$



(a) Dropping probability $\times \sqrt{\lambda}$ (b) Average waiting time $\times \sqrt{\lambda}$ (c) Waiting probability

Fig. 6. The scaling law



(a) Dropping probability (b) Average waiting time (c) Waiting probability

Fig. 7. Sensitivity on α and β

$$Q_3^\lambda \triangleq \left(\sum_{j=0}^{N-K-1} \frac{1}{j!}\left(\frac{K\mu}{\nu}+\frac{K\mu}{\theta}\right)^j \frac{1}{K!}\rho^{N-K-1}\left(\frac{\lambda}{\mu}\right)^K \frac{1}{1-\rho}\right) \times \exp\left(-\left(\frac{\lambda}{\mu}+\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)\right), \quad (10)$$

$$P_1^\lambda \triangleq \left(\sum_{x=0}^{K-1} \frac{1}{x!}\frac{1}{(N-x)!}\left(\frac{\lambda}{\mu}\right)^x \left(\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)^{N-x}\right) \times \exp\left(-\left(\frac{\lambda}{\mu}+\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)\right), \quad (11)$$

$$P_2^\lambda \triangleq \left(\sum_{x=K}^{N} \frac{1}{K!K^{x-K}}\frac{1}{(N-x)!}\left(\frac{\lambda}{\mu}\right)^x \left(\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)^{N-x}\right) \times \exp\left(-\left(\frac{\lambda}{\mu}+\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)\right), \quad (12)$$

$$P_3^\lambda \triangleq \sum_{x=K}^{N}\sum_{0\le y+z\le N-x} \frac{1}{K!}\frac{1}{K^{x-K}}\left(\frac{\lambda}{\mu}\right)^x \frac{1}{y!}\left(\frac{\lambda}{\nu}\right)^y \frac{1}{z!}\left(\frac{\lambda}{\theta}\right)^z \times \exp\left(-\left(\frac{\lambda}{\mu}+\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)\right). \quad (13)$$

$$P_4^\lambda \triangleq \left(\sum_{x=K}^{N} \frac{x-K}{K!K^{x-K}}\left(\frac{\lambda}{\mu}\right)^x \frac{1}{(N-x)!}\left(\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)^{N-x}\right) \times \exp\left(-\left(\frac{\lambda}{\mu}+\frac{\lambda}{\nu}+\frac{\lambda}{\theta}\right)\right), \quad (14)$$

Recalling (1), (2) and applying (3), (4), we obtain,

$$\mathbb{P}[\text{drop}] = \frac{P_1^\lambda + P_2^\lambda}{Q_1^\lambda + Q_2^\lambda - Q_3^\lambda}, \quad (15)$$

$$\mathbb{P}[\text{wait}] = \frac{P_3^\lambda}{Q_1^\lambda + Q_2^\lambda - Q_3^\lambda}, \quad (16)$$

$$\mathbb{E}[W|\text{success}] \sim \frac{P_4^\lambda}{Q_1^\lambda + Q_2^\lambda - Q_3^\lambda}. \quad (17)$$

Now we will compute the limits $Q_i^\lambda, 1 \le i \le 3$ and $P_i^\lambda, 1 \le i \le 4$ as $\lambda \to \infty$. We begin with Q_1^λ. Let X and Y be two independent Poisson random variables with $\mathbb{E}[X] = \lambda/\mu$ and $\mathbb{E}[Y] = \lambda/\nu + \lambda/\theta$.

Therefore, for $\delta_\lambda \triangleq \epsilon\sqrt{\lambda/\mu}$, $\epsilon > 0$ and $m = \lfloor 1/\epsilon^2 \rfloor$, we obtain

$$\begin{aligned} Q_1^\lambda &= \mathbb{P}[X \le K, X+Y \le N] \\ &\le \mathbb{P}[K - m\delta_\lambda \le X \le K, X+Y \le N] \\ &\quad + \mathbb{P}[X < K - m\delta_\lambda] \\ &\le \sum_{i=0}^{m} \Big(\mathbb{P}\left[X \in [K - i\delta_\lambda, K - (i-1)\delta_\lambda]\right] \\ &\quad \times \mathbb{P}\left[Y \le N - (K - i\delta_\lambda)\right]\Big) + \mathbb{P}[X < K - m\delta_\lambda]. \end{aligned} \quad (18)$$

The condition (7) implies

$$\lim_{\lambda\to\infty} \frac{K - i\epsilon\sqrt{\lambda/\mu} - \lambda/\mu}{\sqrt{\lambda/\mu}} = \alpha - i\epsilon,$$

and

$$\lim_{\lambda\to\infty} \frac{N - K + i\epsilon\sqrt{\lambda/\mu} - (\lambda/\nu + \lambda/\theta)}{\sqrt{\lambda/\nu + \lambda/\theta}} = \frac{\beta/\sqrt{\theta} - \alpha/\sqrt{\mu} + i\epsilon/\sqrt{\mu}}{\sqrt{1/\nu + 1/\theta}},$$

which, by the Central Limit Theorem, yields, as $\lambda \to \infty$,

$$\begin{aligned}\mathbb{P}[X < K - i\delta_\lambda] &= \mathbb{P}\left[\frac{X - \lambda/\mu}{\sqrt{\lambda/\mu}} < \frac{K - i\epsilon\sqrt{\lambda/\mu} - \lambda/\mu}{\sqrt{\lambda/\mu}}\right] \\ &\to \Phi\left(\alpha - i\epsilon\right),\end{aligned}$$

as well as

$$\begin{aligned}&\mathbb{P}\left[Y \le N - (K - i\delta_\lambda)\right] = \\ &\mathbb{P}\left[\frac{Y - (\lambda/\nu + \lambda/\theta)}{\sqrt{\lambda/\nu + \lambda/\theta}} < \frac{N - K + i\epsilon\sqrt{\lambda/\mu} - (\lambda/\nu + \lambda/\theta)}{\sqrt{\lambda/\nu + \lambda/\theta}}\right] \\ &\to \Phi\left(\frac{\beta/\sqrt{\theta} - \alpha/\sqrt{\mu} + i\epsilon/\sqrt{\mu}}{\sqrt{1/\nu + 1/\theta}}\right). \qquad (19)\end{aligned}$$

Applying the preceding result to (18) and recalling $m = \lfloor 1/\epsilon^2 \rfloor$, we obtain

$$\begin{aligned}\overline{\lim_{\lambda\to\infty}}\, Q_1^\lambda \le \sum_{i=0}^{\lfloor 1/\epsilon^2 \rfloor} &\left(\Phi\left(\alpha - (i-1)\epsilon\right) - \Phi\left(\alpha - i\epsilon\right)\right) \\ &\times \Phi\left(\frac{\beta/\sqrt{\theta} - \alpha/\sqrt{\mu} + i\epsilon/\sqrt{\mu}}{\sqrt{1/\nu + 1/\theta}}\right) + \Phi\left(\alpha - 1/\epsilon\right),\end{aligned}$$

which, passing $\epsilon \to 0$, yields

$$\begin{aligned}\overline{\lim_{\lambda\to\infty}}\, Q_1^\lambda &\le \int_0^\infty \Phi\left(\frac{\beta/\sqrt{\theta} - \alpha/\sqrt{\mu} + x/\sqrt{\mu}}{\sqrt{1/\nu + 1/\theta}}\right) d\Phi\left(\alpha - x\right) \\ &= \int_{-\infty}^{\alpha} \Phi\left(\frac{\beta/\sqrt{\theta} - x/\sqrt{\mu}}{\sqrt{1/\nu + 1/\theta}}\right) d\Phi\left(x\right). \qquad (20)\end{aligned}$$

On the other hand, we have

$$\begin{aligned}Q_1^\lambda &= \mathbb{P}[X \le K, X + Y \le N] \\ &\ge \sum_{i=0}^{m} \mathbb{P}\left[X \in [K - i\delta_\lambda, K - (i-1)\delta_\lambda]\right] \\ &\quad \times \mathbb{P}\left[Y \le N - (K - (i-1)\delta_\lambda)\right],\end{aligned}$$

which, following the same approaching in computing (20), results in

$$\underline{\lim_{\lambda\to\infty}}\, Q_1^\lambda \ge \int_{-\infty}^{\alpha} \Phi\left(\frac{\beta/\sqrt{\theta} - x/\sqrt{\mu}}{\sqrt{1/\nu + 1/\theta}}\right) d\Phi\left(x\right). \qquad (21)$$

Combining (20) and (21) proves that $\lim_{\lambda\to\infty} Q_1^\lambda = Q_1$.

Next, we compute Q_2^λ. Recalling Y defined before (18) and $\rho = \lambda/(K\mu)$, and using Stirling's formula

$$K! \sim \sqrt{2\pi K} K^K e^{-K},$$

we obtain, as $\lambda \to \infty$,

$$\begin{aligned}Q_2^\lambda &= \left(\sum_{j=0}^{N-K-1} \frac{1}{j!}\left(\frac{\lambda}{\mu} + \frac{\lambda}{\theta}\right)^j\right) \exp\left(-\left(\frac{\lambda}{\nu} + \frac{\lambda}{\theta}\right)\right) \\ &\quad \times \frac{1}{K!}\frac{1}{K}\left(\frac{\lambda}{\mu}\right)^{K+1} \frac{e^{-K\rho}}{1-\rho} \\ &\sim \mathbb{P}[Y \le N - K - 1] \times \frac{e^K}{\sqrt{2\pi K} K^{K+1}} \left(K\rho\right)^{K+1} \\ &\quad \times \sqrt{\frac{K}{\rho}} \frac{1}{\alpha} e^{-K\rho} \\ &\sim \frac{e^{K(1-\rho+\log\rho)}}{\sqrt{2\pi}\alpha} \Phi\left(\frac{-\alpha\sqrt{1/\nu} + \beta\sqrt{1/\theta}}{\sqrt{1/\nu + 1/\theta}}\right),\end{aligned}$$

which, using $1 - \rho \sim \alpha\sqrt{\rho/K}$ and $\log\rho = (1-\rho) - (1-\rho)^2 + o((1-\rho)^2)$, yields

$$\lim_{\lambda\to\infty} Q_2^\lambda = \frac{e^{-\alpha^2/2}}{\sqrt{2\pi}\alpha} \Phi\left(\frac{-\alpha\sqrt{1/\nu} + \beta\sqrt{1/\theta}}{\sqrt{1/\nu + 1/\theta}}\right).$$

Following the same approach in computing Q_2^λ, we can prove that $\lim_{\lambda\to\infty} Q_3^\lambda = Q_3$. Due to the limited space, we omit the details.

Then, we focus on P_1^λ. For a Poisson point process of unit rate on $[0, \lambda/\mu + \lambda/\nu + \lambda/\theta]$, denote by X the number of points on $[0, \lambda/\mu)$ and Y on $[\lambda/\mu, \lambda/\mu + \lambda/\nu + \lambda/\theta]$. It is clear that X and Y follow Poisson distributions of rate λ/μ and $\lambda/\nu + \lambda/\theta$, respectively. Similar to Q_1^λ, we obtain

$$\begin{aligned}P_1^\lambda &= \mathbb{P}[X < K, X + Y = N] \\ &= \mathbb{P}[X < K | X + Y = N]\mathbb{P}[X + Y = N]. \qquad (22)\end{aligned}$$

It is well know that conditional on $X + Y = N$, each point on $[0, \lambda/\mu + \lambda/\nu + \lambda/\theta]$ is uniformly distributed by the property of Poisson processes. Therefore, for an i.i.d. Bernoulli sequence $Z_i, i \ge 1$ with $\mathbb{E}[Z_i] = \mathbb{E}[X]/(\mathbb{E}[X+Y]) = (1/\mu)/(1/\mu + 1/\nu + 1/\theta) \triangleq \delta$ and $\mathbb{V}[Z_i] = (1/\mu)(1/\nu + 1/\theta)/(1/\mu + 1/\nu + 1/\theta)^2 = \delta(1-\delta)$, we have

$$\begin{aligned}\mathbb{P}[X < K | X + Y = N] &= \mathbb{P}\left[\sum_{i=1}^{N} Z_i < K\right] \\ &= \mathbb{P}\left[\frac{\sum_{i=1}^N Z_i - N\delta}{\sqrt{N\delta(1-\delta)}} < \frac{K - N\delta}{\sqrt{N\delta(1-\delta)}}\right],\end{aligned}$$

which, by the Central Limit Theorem, yields

$$\begin{aligned}&\lim_{\lambda\to\infty} \mathbb{P}[X < K | X + Y = N] \\ &\qquad = \Phi\left(\frac{\alpha/\sqrt{\mu} - \beta\delta/\sqrt{\theta}}{\sqrt{\left(1/\mu + 1/\nu + 1/\theta\right)\delta(1-\delta)}}\right). \qquad (23)\end{aligned}$$

Regarding $\mathbb{P}[X + Y = N]$, we can show that

$$\lim_{\lambda\to\infty} \sqrt{\lambda}\mathbb{P}[X + Y = N] = \sqrt{\mu\delta}\varphi\left(\frac{\sqrt{2}\beta}{\sqrt{\theta/\mu + \theta/\nu + 1}}\right),$$

which, in conjunction with (23), proves $\lim_{\lambda\to\infty} P_1^\lambda = P_1$.

Next we compute P_2^λ. Defining a random variable W with $\mathbb{P}[W=n]=(1-\rho)\rho^n/\rho^K$, $n=K,K+1,\cdots$. We obtain, recalling the definition of Y,

$$P_2^\lambda = \frac{K^K}{K!}\frac{\rho^K}{1-\rho}e^{-K\rho}\left(\sum_{x=K}^{N}\mathbb{P}[W=x]\mathbb{P}[Y=N-x]\right).$$

Using the fact that $\mathbb{P}[W=x]$ is a monotonically decreasing function with respect to x and

$$\frac{K^K}{K!}\frac{\rho^{K+1}}{1-\rho}e^{-K\rho}\to\frac{\varphi(\alpha)}{\alpha},\text{ as }\lambda\to\infty,$$

we have, for $\delta_\lambda=\epsilon\sqrt{\lambda/\mu}$ and $m=\lfloor 1/\epsilon^2\rfloor$,

$$P_2^\lambda \gtrsim \frac{\varphi(\alpha)}{\alpha}\Bigg(\sum_{i=1}^{m}\mathbb{P}[W=K+i\delta_\lambda] \mathbb{P}\left[N-(K+(i+1)\delta_\lambda)\le Y<N-(K+i\delta_\lambda)\right]\Bigg).$$

It can be shown that

$$\lim_{\lambda\to\infty}\sqrt{\lambda}\mathbb{P}[W=K+i\delta_\lambda]=\alpha\sqrt{\mu}e^{-\alpha i\epsilon}, \tag{24}$$

and (19) implies

$$\begin{aligned}\lim_{\lambda\to\infty}&\mathbb{P}\left[N-(K+(i+1)\delta_\lambda)\le Y<N-(K+i\delta_\lambda)\right]\\ &=\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}-i\epsilon/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)\\ &\quad-\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}-(i+1)\epsilon/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right).\end{aligned} \tag{25}$$

Combining (24), (25) and passing $\epsilon\to 0$, we obtain

$$\begin{aligned}&\underline{\lim}_{\lambda\to\infty}\sqrt{\lambda}P_2^\lambda\ge\\ &\int_0^\infty\varphi(\alpha)\sqrt{\mu}e^{-\alpha x}d\Phi\left(\frac{\beta/\sqrt{\theta}-(\alpha+x)/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)\\ &=\varphi(\alpha)\sqrt{\mu}\exp\left(\alpha^2/\delta-\alpha\beta\sqrt{\frac{\mu}{\theta}}\right)\\ &\quad\times\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}-\alpha\sqrt{\frac{\mu}{\nu}+\frac{\mu}{\theta}}\right).\end{aligned} \tag{26}$$

Using the monotonicity of $\mathbb{P}[W=x]$ we can also compute $\overline{\lim}_{\lambda\to\infty}\sqrt{\lambda}P_2^\lambda$, which coincides with $\overline{\lim}_{\lambda\to\infty}\sqrt{\lambda}P_2^\lambda$.

Following the same approach in computing P_2^λ, we obtain

$$P_3^\lambda\gtrsim\frac{\varphi(\alpha)}{\alpha}\Bigg(\sum_{i=1}^{m}\mathbb{P}\left[K+i\delta_\lambda\le W<K+(i+1)\delta_\lambda\right]\mathbb{P}\left[Y<N-(K+i\delta_\lambda)\right]\Bigg)$$

$$\begin{aligned}&\sim\frac{\varphi(\alpha)}{\alpha}\sum_{i=1}^{m}\left(e^{-\alpha i\epsilon}-e^{-\alpha(i+1)\epsilon}\right)\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}-i\epsilon\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)\\ &\to\varphi(\alpha)\int_0^\infty\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}-x\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)e^{-\alpha x}dx\\ &=\frac{\varphi(\alpha)}{\alpha}\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)-\frac{\varphi(\alpha)}{\alpha}\exp\left(\alpha^2/\delta-\alpha\beta\sqrt{\frac{\mu}{\theta}}\right)\\ &\quad\times\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}-\alpha\sqrt{\frac{\mu}{\nu}+\frac{\mu}{\theta}}\right).\end{aligned} \tag{27}$$

Similarly we can show that $\overline{\lim}_{\lambda\to\infty}P_3^\lambda\le P_3$. Using the same approach as in calculating P_3^λ, we can prove that $\lim_{\lambda\to\infty}\sqrt{\lambda}P_4^\lambda$ is equal to

$$\frac{\varphi(\alpha)}{\sqrt{\mu}}\int_0^\infty xe^{-\alpha x}\Phi\left(\frac{\beta/\sqrt{\theta}-\alpha/\sqrt{\mu}-x\sqrt{\mu}}{\sqrt{1/\nu+1/\theta}}\right)dx.$$

Applying all these computed limits in (15), (16) and (17) finishes the proof of the theorem.

REFERENCES

[1] Kernel based virtual machine. http://www.linux-kvm.org/page/Main_Page

[2] IBM Scale Out Network Attached Storage. http://www-03.ibm.com/systems/storage/network/sonas/

[3] A. Anandkumar, C. Bisdikian, and D. Agrawal. Tracking in a spaghetti bowl: monitoring transactions using footprints. In *Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '08, pages 133–144, New York, NY, USA, 2008. ACM.

[4] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005.

[5] A. Benveniste, E. Fabre, S. Haar, and C. Jard. Diagnosis of asynchronous discrete-event systems: A net unfolding approach. *IEEE Transactions on Automatic Control*, 48:2003, 2003.

[6] S. C. Borst, A. Mandelbaum, and M. I. Reiman. Dimensioning large call centers. *Operations Research*, 52:17–34, 2000.

[7] R. N. Calheiros, R. Ranjany, and R. Buyya. Virtual machine provisioning based on analytical performance and qos in cloud computing environments. In *International Conference on Parallel Processing (ICPP)*, 2011.

[8] H. Chen and D. D. Yao. *Fundamentals of Queueing Networks*. Springer, June 2001.

[9] A. P. Estrada-Vargas, E. López-Mellado, and J.-J. Lesage. Off-line identification of concurrent discrete event systems exhibiting cyclic behaviour. In *Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, SMC'09, pages 181–186, Piscataway, NJ, USA, 2009. IEEE Press.

[10] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29(3):567–588, 1981.

[11] P. Khudyakov, P. D. Feigin, and A. Mandelbaum. Designing a call center with an IVR (interactive voice response). *Queueing Syst.*, 66(3):215–237, 2010.

[12] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proceedings of the 30th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2011)*, pages 1098–1106, Shanghai, China, April 2011.

[13] A. Quiroz, H. Kim, M. Parashar, N. Gnanasambandam, and N. Sharma. Towards autonomic workload provisioning for enterprise grids and clouds. In *IEEE/ACM International Conference on Grid Computing (GRID)*, 2009.

[14] G. Wang and T. Ng. The impact of virtualization on network performance of Amazon EC2 data center. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, March 2010.

[15] S. Zeltyn and A. Mandelbaum. Call centers with impatient customers: Many-server asymptotics of the M/M/n + G queue. *Queueing Systems: Theory and Applications*, 51:361–402, December 2005.