Multiobjective Cloud Capacity Planning for Time-Varying Customer Demand

Brian Bouterse  
Department of Computer Science  
North Carolina State University  
Raleigh, NC, USA  
bmbouter@ncsu.edu

Harry Perros  
Department of Computer Science  
North Carolina State University  
Raleigh, NC, USA  
hp@ncsu.edu

David Thuente  
Department of Computer Science  
North Carolina State University  
Raleigh, NC USA  
thuente@ncsu.edu

Abstract—Service providers who dynamically scale cloud resources can significantly lower costs while providing a service level that conforms to a service level agreement. To do so, service providers must understand the tradeoffs within provisioning algorithms such as utilization versus system availability or the impact of service level agreement timescale on utilization. Within the context of three provisioning algorithms from existing literature, we analyze the tradeoff of service availability versus utilization using a two-dimensional Pareto analysis. We also analyze the impact on utilization and system availability of using hourly, daily, weekly, or yearlong timescales as the basis of service level agreement. We evaluate model performance using historical data of the Virtual Computing Laboratory (VCL), a cloud computing environment at North Carolina State University. We show that a simple heuristic planning model, whereby a fixed reserve capacity is maintained, provides better service availability and utilization performance than other models for all service level timescales. The fixed reserve capacity model is also shown to be Pareto optimal.

Keywords—capacity planning, auto scaling, application delivery, VCL, Pareto, service level agreement timescale, virtualization, non-stationary traffic, non-homogeneous traffic

I. INTRODUCTION

This paper examines the problem of estimating the number of virtual machines that a Software-as-a-Service provider should maintain to satisfy a 99th percentile Service Level Agreement (SLA) measured on hourly, daily and weekly timescales to support a compute-intensive desktop application to customers. The impact of SLA timescale on utilization service performance is also examined. The customer demand varies as a function of time. Three models from existing literature [1] are compared for estimating the required number of virtual machines, so that the customer blocking probability as measured on a given timescale is low and the utilization is high. Historical data from the Virtual Computing Laboratory (VCL) at North Carolina State University [2] is used for this evaluation. The VCL is a cloud computing platform consisting of around 2,500 blades, and it offers computing services to 42,000 students and faculty who use it for teaching and research. As a Software-as-a-Service provider, the VCL contains the right challenges to practically benefit from a schedule of bringing online and offline cloud computing resources that maintains service availability and maximizes utilization.

Software-as-a-Service providers are faced with a multi-objective optimization problem whereby blocking probability should be minimized and utilization should be maximized. Under-provisioning increases the blocking probability while at the same time it increases the utilization as more end-users compete for a smaller number of application seats. On the other hand, overprovisioning results in wasted resources causing a decrease in utilization and in the blocking probability. The conflicting behavior of blocking probability minimization versus utilization optimization create a Pareto optimization problem, whereby multiple solutions are considered equally valid optimizations in the absence of subjective information relating the importance of optimizing a specific dimension. In this paper, the Pareto optimal points of each model are identified, but in addition, it is assumed as an example that a Software-as-a-Service provider subjectively prefers maximizing utilization while maintaining a blocking probability less than or equal to 1%.

The client side provisioning models explored in this paper have had their blocking probability performance reported within existing literature [1]. Similar models are also reported by Groskinsky et al. [3] who created models which target a blocking probability goal within the range 0.01 – 0.03. This paper evaluates blocking probability performance, but does not assume a preferred service level by the service provider and explores both service level performance and utilization as a two-dimensional optimization problem. Many provisioning studies have been done that focus on dynamic cloud provisioning that adheres to a service level agreement [4], [5], [6], but we do not know of a dynamic cloud provisioning study that examines the impact of reducing the timescale of performance assessment. Martin and Nilsson [7] identify the impact on WAN performance of shortening the performance reporting timescale.

The paper is organized as follows. Section 2 presents the system under study. Section 3 presents and analyzes a very simple model referred to as the fixed capacity model. Section 4 presents and evaluates a model based on an exponential moving average traffic predictor and an Erlang-loss capacity planning technique. In section 5, a simple heuristic model is
presented where a fixed reserve capacity is maintained at each five-minute interval. All models and their performance are expressed in terms of the blocking probability and the utilization evaluated on an hourly, daily, weekly, and yearlong timescales. Section 6 compares the above discussed models using a Pareto analysis through which performance optimality points across all models are identified. Finally, the conclusions are given in section 7.

II. THE SYSTEM UNDER STUDY

It is assumed that a Software-as-a-Service provider dynamically requests virtual machines from an Infrastructure-as-a-Service cloud provider. Each virtual machine contains the software application of interest and has \( N \) seats, that is, each virtual machine can serve \( N \) different end-users concurrently. In the experiments presented in this paper, we assume that \( N=2 \), an assumption resulting from the VCL that allows two users per virtual machine. The Software-as-a-Service provider, VCL, requests virtual machines dynamically as customer demand increases, and releases them dynamically as demand decreases. The increases and decreases of servers are in integer numbers. If an end-user arrives at a time when there are no seats available, the end-user is blocked and leaves the system without returning. An infinite population of end-users is assumed, and also it is assumed that the Infrastructure-as-a-Service provider can provide as many virtual machines as requested by the Software-as-a-Service provider.

![Number of Requests per Five-Minute Interval for the Entire Year](image1.png)

Fig. 1. Number of Requests per Five-Minute Interval for the Entire Year

The proposed models are tested using the historical data from VCL collected over a year, starting on July 1. End-users that have a service time longer than 8 hours are removed from the data set. This is because the maximum on-demand reservation in VCL for normal end-users is 8 hours. Longer reservations are also accepted but they are provisioned in a dedicated way, which is not within the scope of this study. The resultant data set consists of a total of 175,554 end-user sessions. Each end-user session is associated with the time at which the end-user arrived and the total service time used, i.e., the total time the end-user held a seat. The number of sessions computed over five-minute intervals for the entire year is shown in Fig. 1, and the histogram of the service times is shown in Fig. 2.

![Histogram of Service Times](image2.png)

Fig. 2. The Service Time Distribution of the Entire Year

As can be seen in Fig. 1, the arrival rate varies daily and weekly. It also varies seasonally as the school calendar transitions through the fall and spring semesters, summer sessions, and holiday periods. For instance, the midpoint in Fig. 1 has very low arrival rate, which corresponds with the academic December holiday. The figure has been annotated for anecdotal insight into the relationship between demand and the academic calendar.

A delay of five minutes between the time that one or more virtual machines are requested to the time they become available is assumed based on a multi-provider study of virtual machine startup time within Infrastructure-as-a-Service environments, see Mao and Humphrey [8]. The authors also identify that the number of requested virtual machines to boot does not impact the virtual machine boot time. In this paper, the same assumption is made, i.e., that the boot time is independent of the requested number of virtual machines. A shut down delay is also introduced in order to avoid “thrashing”, see Groskinsky et al. [3], whereby a virtual machine is released and immediately after a new virtual machine is requested. Such an oscillation can cause an unnecessary boot time. The shut down time is also set to five minutes. After five minutes, a released virtual machine is returned to the Infrastructure-as-a-Service provider and it is no longer part of the pool of virtual machines used by the Software-as-a-Service provider. When a new virtual machine is requested, released virtual machines that have not been returned to the Infrastructure-as-a-Service provider yet are first examined. If there is one available then the virtual machine is put into service without incurring a boot time. In addition, a virtual machine can only be released from active service when all the seats of the virtual machine are empty.

Three models are considered in this paper for provisioning the capacity of the above system. The performance of each model is expressed in terms of the blocking probability and server utilization. The blocking probability is defined as the probability that a request is denied because there are no seats available. The server utilization is defined as the proportion of seat time that is used by end-users. The blocking probability is expressed as the 99th percentile of observed values, which are calculated for the following timescales: hourly, daily, and weekly. In addition the average blocking probability over the entire year is calculated. As will be seen, the annual blocking probability provides a lower bound on the required capacity. Utilization is reported as an average, since it is a service provider metric to evaluate infrastructure efficiency. All results given in this paper are for an entire year, and consequently, the
average utilization is the same for all timescales and coincides with the annual average. The 99th percentile of the utilization is also calculated to match with the 99th percentile of the blocking probability, but it is not deemed to be a meaningful metric. To calculate the blocking probability and the utilization of each model, a simulation was implemented in SimPy [9], which depicts the system described above with the provisioning of the capacity obtained by the model under test.

III. THE FIXED CAPACITY MODEL

This is a simple model where the number of virtual machines required is fixed, that is, it does not vary over time. The total number of seats, s, is computed using the historical demand data from VCL, as follows. Using the simulation model, the blocking probability and utilization for the historical data is calculated for various values of the total seat capacity s assuming N=2. The startup delay and shutdown delay in the simulation model are set to zero because s is fixed at the beginning, and the number of virtual machines provisioned is never changed. The 99th percentile blocking probability per hour, day, and week over the entire year of data, along with the average blocking probability and utilization for various values of s as shown in Fig. 3.

As shown in Fig. 3, each explored timescale is Pareto optimal, indicating that there is no alternate value of s that simultaneously increases utilization while decreasing blocking probability within a specific timescale. Therefore, without additional subjective preferences, all s values are considered equally optimal. Using the subjective preference of maximizing utilization while keeping the blocking probability to less than or equal to 1%, the optimal points for the weekly timescale is s=168, i.e., 84 virtual machines, which produces a utilization of 0.16441.

As shown in Fig. 3, each explored timescale is Pareto optimal, indicating that there is no alternate value of s that simultaneously increases utilization while decreasing blocking probability within a specific timescale. Therefore, without additional subjective preferences, all s values are considered equally optimal. Using the subjective preference of maximizing utilization while keeping the blocking probability to less than or equal to 1%, the optimal points for the weekly timescale is s=168, i.e., 84 virtual machines, which produces a utilization of 0.16441.

![Graph showing utilization and blocking probability for different timescales](image)

**Fig. 3.** Fixed Capacity Pareto Efficiency with Utilization and 99th Percentile Blocking Probability for the Weekly, Daily, and Hourly Timescales and Yearlong Mean vs s

Fig. 3 compares the 99th percentile blocking probability and average utilization across the weekly, daily, and hourly timescales along with the yearlong mean. Overall, as timescales get shorter, the same values of s produce blocking probability increases. The only exception is with high values of s, which result in the hourly timescale producing the lowest 99th percentile of the blocking probability and the highest utilization. Optimality points of shorter timescales become suboptimal when compared to a longer timescale, indicating that a larger capacity is required to produce the same blocking probability for shorter timescales. Generally, performance is ranked from best to worst with annual, weekly, daily, and hourly respectively with s < 116, but for s >= 116 the hourly timescale is the highest performing.

The Pareto optimality points identify model performance that best optimize both service availability and utilization. The annual timescale is given as a lower bound for comparison; of the weekly, daily, and hourly timescales the weekly timescale occupies the Pareto optimality points for s < 116, but the hourly timescale occupies the Pareto points for s >= 116.

IV. THE EXPONENTIAL MOVING AVERAGE MODEL

The exponential moving average model considered in this paper is described in Bouterse and Perros [1]. It predicts the number of customer arrivals during the next five-minute interval. The model is as follows:

\[
s_t = x_0 \\
s_t = \alpha x_{t-1} + (1-\alpha)s_{t-1} = s_{t-1} + \alpha(x_{t-1} - s_{t-1}), \quad t > 1
\]

(1)

where \(x_t\) is the observed number of customers who arrived during the \(t^{th}\) five-minute interval, and \(s_t\) is the exponential moving average predicted number of customers who will arrive during the \(t^{th}\) five-minute interval. For each five-minute interval, a prediction of the number of arrivals is made using the exponential moving average equation (1). The number of arrivals is used as the offered traffic load to an Erlang loss model, as described by Bouterse and Perros [1], to make a seat capacity prediction for that time interval.

The exponential moving average is parameterized by the constant coefficient \(\alpha\), where \(0 \leq \alpha \leq 1\). This model was used in the simulation for various values of \(\alpha\) for historical VCL data and the 99th percentile of the blocking probability is obtained for hourly, daily, and weekly timescales, along with the average utilization and the yearlong average blocking probability. As before, \(N=2\) seats per virtual machine is assumed, and boot time and shut down times equal to five minutes.

As shown in Fig. 4, on all explored timescales the exponential moving average model has non Pareto optimal points, indicating that there are many values of \(\alpha\) that are suboptimal and increase blocking probability while also lowering utilization relative to other choices of \(\alpha\). Overall the model performs poorly, with lower utilization and higher blocking probability than other models evaluated in [1]. model performance becomes disjoint and non-continuous with small changes to \(\alpha\) on shorter timescales, i.e. the hourly, daily, and weekly timescales, but becomes continuous for the average blocking probability calculated over the year. The yearlong average blocking probability and utilization is given as a lower bound of performance as timescales are lengthened.
A simple heuristic for determining the capacity of the VCL was proposed in [1], whereby a fixed number $R$ of unused seats is continuously maintained. Specifically, every five minutes, the number of in-use seats, $I$, and unused seats, $U$, are observed and accordingly the total capacity of the system, expressed in seats, $s$ is set to:

$$s = I + \left\lceil \frac{R}{N} \right\rceil \times N \quad (2)$$

Ideally, the unused available seats would be set exactly to $R$, but virtual machines have a seat capacity, $N$, forcing total seat capacity, $s$, to be an even multiple of $N$. This model is referred to as the reserve capacity model. The reserve capacity model uses $R$ to change the capacity every five-minute period, but $R$ is fixed at the beginning and it is not varied with time. A parameterized search for optimized value of $R$ was carried out using the historical VCL data. For each value of $R$, the $99^\text{th}$ percentile of the blocking probability for the hourly, daily, and weekly timescales along with the average blocking probability and utilization for the entire year are shown in Fig. 5. As assumed above, a seat density of $N=2$ seats per virtual machine are assumed, along with boot time and shut down times equal to five minutes.

Fig. 5 gives a comparison of the $99^\text{th}$ percentile of the blocking probability of the weekly, daily, and hourly timescales and the yearlong mean versus average, annual utilization. Overall, as timescales get shorter, the same values of $R$ cause blocking probability increases, with the exception of high values of $R$, which result in the hourly timescale producing the highest $99^\text{th}$ percentile of the blocking probability. Of the weekly, daily, and hourly timescales, performance is ranked from best to worst with weekly, daily, and hourly respectively with $R < 16$, but for $R >= 16$ the hourly timescale is the best performing.

The reserve capacity model is Pareto optimal within each explored timescale indicating that for any given timescale there is no alternate value of $R$ that simultaneously increases utilization while decreasing blocking probability relative to any other value of $R$. Therefore, without additional subjective preferences, all points within a timescale are considered equally optimal. The yearlong average blocking probability and utilization is given as a lower bound of performance as timescales are lengthened.

Using the subjective preference of maximizing utilization while keeping the blocking probability to less than or equal to 1%, optimality points exist on all blocking probability timescales, and are shown in Table 1.

An interesting outcome identified in Table 1 is that to adhere to a subjective preference of a $99^\text{th}$ percentile of blocking probability that is less than or equal to 0.01, the daily and weekly timescales require an $R$ value that is roughly twice
as large as the hourly timescale. This is a trend that can also be seen in Fig. 5. The lowest 99th percentile of the blocking probability on the hourly timescale occurs when $R \geq 16$, and given that the subjective blocking probability of interest falls within the $R \geq 16$ region, the hourly timescale produces a better performance than the daily or weekly timescales with a lower $R$. The yearlong blocking probability is given as a lower bound of performance, and produces the lowest blocking probability and highest utilization of any timescale with $R=12$.

### VI. Model Comparison

A comparison of blocking probability and utilization outcomes of the models is done using two-dimensional analysis where the 99th percentile of the blocking probability within a weekly timescale and the average annual utilization are plotted in Fig. 6.

![Comparison of Models for the Weekly Timescale](image)

Fig. 6 shows that the reserve capacity model occupies the entire Pareto optimality points indicating it provides better performance in all cases independent of the subjective preferences the service provider has about the relative importance of utilization versus service availability. The other timescales are not included in this paper, but the reserve capacity similarly outperforms all other models on the hourly, daily, and yearlong timescales as well.

When evaluated using the subjective preference of achieving a blocking probability of less than or equal to 0.01, while maximizing utilization, the reserve capacity model is able to maintain service level goals while maximizing utilization on all timescales explored with different values of $R$. Optimized values of $R$ for different timescales can be seen in Table 4.6 above, which identifies the optimality points of the subjective preferences across all models.

### VII. Conclusions

The reserve capacity model outperforms both the fixed capacity model and exponential moving average model by providing significantly higher utilization for equivalent blocking probabilities for all examined timescales. The reserve capacity model also has values of $R$ that satisfy the subjective criteria of 0.01 blocking probability with the highest utilization on all timescales.

An intuitive outcome is observed across all models whereby shorter timescales result in higher blocking probabilities, indicating that shorter timescales generally require more resources to achieve similar blocking probability performance.

One interesting outcome related to timescales is that for the fixed capacity and reserve capacity models the hourly timescale outperforms the other timescales in regions with low blocking probabilities. This is likely due to the hourly timescale having more SLA periods relative to other timescales, which causes hours with SLA violations to be drowned out 99th percentile being computed from a large number of hourly samples which lowers the, effectively lowering the blocking probability as measured by the service level agreement. This can also be observed in the amount of reserve capacity required to support a blocking probability of less than 0.01; at that service level the hourly timescale requires roughly half as much reserve capacity as the daily or weekly timescales. This outcome has implications on the viability of short timescales for service level agreement for service providers with high service levels.

### REFERENCES


