Regression-based resource provisioning for session slowdown guarantee in multi-tier Internet servers

Sireesha Muppala, Xiaobo Zhou, Liqiang Zhang, Guihai Chen

A R T I C L E   I N F O

Article history:
Received 2 October 2010
Received in revised form 16 August 2011
Accepted 29 November 2011
Available online 8 December 2011

Keywords:
Scalable multi-tier Internet services
Autonomous resource provisioning
Performance guarantee
Session slowdown
Regression modeling and analysis

A B S T R A C T

Autonomous management of a multi-tier Internet service involves two critical and challenging tasks, one understanding its dynamic behaviors when subjected to dynamic workloads and second adaptive management of its resources to achieve performance guarantees. We propose a statistical machine learning based approach to achieve session slowdown guarantees of a multi-tier Internet service. Session slowdown is the relative ratio of a session’s total queueing delay to its total processing time. It is a compelling performance metric of session-based online transactions because it directly measures user-perceived relative performance and it is independent of the session length. However, there is no analytical model for session slowdown on multi-tier servers. We first conduct training to learn the statistical regression models that quantitatively capture an Internet service’s dynamic behaviors as relationships between various service parameters. Then, we propose a dynamic resource provisioning approach that utilizes the learned regression models to efficiently achieve session slowdown guarantee under dynamic workloads. The approach is based on the combination of offline training and online monitoring of the Internet service behavior. Simulations using the industry standard TPC-W benchmark demonstrate the effectiveness and efficiency of the regression based resource provisioning approach for session slowdown oriented performance guarantee of a multi-tier e-commerce application.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Multi-tier architecture is becoming a standard for modern Internet applications. Many popular Internet applications employ a multi-tier architecture with each tier depending on its successor and providing functionality to its preceding tier [1,2,4,7,12–16, 19–22,29,31–34,40]. The multi-tier systems continue to grow in scale and complexity. They become so complicated that it is even a big challenge to get a good understanding of the entire system dynamic behavior [2,25]. Providing performance guarantee to meet service level agreements for complex multi-tier Internet services is a critical but challenging task.

Typical Internet workloads are session based. A session is a sequence of individual requests of different types made by a customer during a single visit to a web site [18]. Resource management techniques have been proposed for request based performance guarantee in multi-tier servers [7,14,31,40]. However, there is a lack of techniques that provide session based performance guarantee. Response time and queueing delay are the major absolute performance metrics in evaluating request based system responsiveness and service quality. However, they are not only unsuitable for comparing requests that have very different resource demands, but also not applicable for session based workloads because of the dynamic session length. Session length, which is the number of requests in a session, is dynamic and unknown at the time of session origination. Because of the unpredictability of the session length, one cannot guarantee the absolute response time of a session. A relative metric independent of session length is favorable and mandatory for performance guarantee of session based Internet services [43].

We promote the use of session slowdown for performance guarantee in multi-tier servers. Slowdown is the relative ratio of a request’s queueing delay to its actual processing time [10,11]. It is known that clients are more likely to anticipate short delays for “small” requests like browsing, and more willing to tolerate long delays for “large” requests like search [43]. Using slowdown as the QoS metric facilitates attaining the anticipation. Because the slowdown metric directly translates to user-perceived relative performance and system load, it has been accepted as an important performance metric on servers [10,11,26,41–43]. Session slowdown is the relative ratio of the total queueing delay of requests of one session to the total processing time of the requests.
It is compelling for session-based performance measurement because it is user-perceived service quality at the session level and it is independent of the session length [43]. However, providing session slowdown guarantee on multi-tier servers is important, but also challenging.

There are queueing analytical models for request slowdown differentiation on single-tier Internet servers [41–43]. Zhou et al. derived a closed form expression of the expected request slowdown in an M/G/1 queue with a bounded Pareto service time distribution [42]. However, the extension of the analytical model to the multi-tier architecture is greatly challenging, if not impossible due to the inter-tier dependences, per-tier replication and caching policies and concurrency limits [7]. An extended model, even if feasible, may not accurately capture the dynamics of session-based workloads on multi-tier servers, resulting in performance guarantee violations.

In this paper, we propose to use a statistical learning based approach to tackle the challenges of modeling multi-tier Internet service dynamics and providing session slowdown guarantee. In a multi-tier Internet service, the resource demand posed by user sessions on different tiers is dynamic in nature leading to the bottleneck tier shift issue [2,19,21]. With an analytical model based approach, dynamic bottleneck tier shift and unpredictable Internet workload timing may lead to resource over-provisioning or service level agreement violations. As multi-tier systems grow in complexity, empirical models built using statistical learning have great potential in overcoming the scalability and complexity challenges [2,4,21,25]. To our best knowledge, our approach should be the first one in the application of the regression principle to session slowdown guarantee on multi-tier servers.

Our main contributions are using statistical regression models to effectively capture the dynamic behavior of a multi-tier Internet service under dynamic and complex session-based workloads, and, the design of a novel regression based dynamic resource provisioning strategy that utilizes learned regression models to provide session slowdown guarantees for the multi-tier service. The approach is based on the combination of offline training and online monitoring of the multi-tier service behaviors. Offline training is conducted using a set of training workloads, with each workload being a single traffic mix arriving at constant session arrival rate. Importantly, we propose to use two distinct statistical regression models to control the upper and lower bounds of resources allocated to the multi-tier Internet service.

We evaluate the effectiveness and efficiency of the proposed approach using the industry standard TPC-W benchmark workloads in a typical 3-tier e-commerce environment. For evaluation, we use realistic workloads consisting of combination of different traffic mixes with dynamically varying session arrival rates. Simulation results demonstrate that the approach adapts to workload variations when it is subjected to a workload different than the training workloads. It achieves session slowdown guarantee for various dynamic workloads while resources are efficiently utilized.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the rationale of using statistical regression analysis. Section 4 gives the statistical learning based dynamic resource provisioning approach for session slowdown guarantee. Section 5 presents experimental results and performance evaluation. Concluding remarks are given in Section 6.

2. Related work

Resource management for quality-of-service provisioning in multi-tier Internet applications is a very important and active research topic [1,2,6–9,13–16,23,27,28,30–33,35–38]. A few studies focused on the modeling and analysis of multi-tier servers with queueing foundations. For instance, Diao et al. described a performance model for differentiated services of multi-tier applications [7]. They addressed per-tier concurrency limits and cross-tier interactions using a M/M/1 queueing model. Villela et al. [33] designed queueing-theoretic methods to provision servers in the application tier with a profit optimization model. Ur-gaonkar et al. [31] proposed an analytic model for session-based multi-tier applications using a network of queues. The mean–value analysis algorithm for queueing networks was used to measure the mean response time. The authors further designed a dynamic provisioning technique on multi-tier server clusters for requests’ average end-to-end delay guarantee [32]. In [14], Lama and Zhou proposed an efficient server provisioning scheme based on an end-to-end resource allocation optimization model and a model-independent fuzzy controller for the average and 90th-percentile request delay guarantees. They designed the model-independent fuzzy controller to address the lack of an accurate performance model.

In a multi-tier system, the user perceived performance is the result of a complex interaction of complex workloads in a very complex underlying system [19]. Workload flows to a data center are often characterized as bursty. Recent studies [19,20,27] have seen highly dynamic workloads of Internet applications that fluctuate over multiple time scales, which can have a significant impact on the processing demands imposed on servers. Mi et al. introduced a new methodology for injecting burstiness to the TPC-W benchmark by changing the user think time based on index of dispersion [20]. Caniff et al. in [3] presented Fastrack, a self-adaptive, parameter-free algorithm that detects the start and end of bursty workload periods using the index of dispersion. In this paper, we injected burstiness to workloads for performance evaluation of the regression based resource provisioning approach for session slowdown guarantee.

Statistical machine learning techniques have been used to measure the capacity of Internet Web sites [4,2,39] and admission control [21]. Chen et al. [4] applies the K-nearest-neighbors (KNN) machine learning approach for adding database replicas in dynamic content Web server clusters. Rao and Xu [25] uses a Bayesian network to correlate low level instrumentation data including system and user CPU time, available memory size, and I/O status that are collected at run-time with high level system states in each tier of a multi-tier web site. A decision tree is induced over a group of coordinated Bayesian models in different tiers to identify the bottleneck dynamically when the system is overloaded. Muppala and Zhou [21] proposed a coordinated session based admission control approach. It uses a Bayesian network to correlate the states of all tiers of an e-commerce application. The probability with which a session is admitted is determined by the probabilistic inference of the network after applying the evidence in terms of utilization and processing time at each tier to the network. Results demonstrate significant performance improvement in effective session throughput.

Statistical learning techniques are also effectively used in configuration and tuning of system parameters. Bu et al. [2] used a reinforcement learning approach for autonomic configuration and reconfiguration of multi-tier web systems. The proposed technique effectively adapts the performance parameter settings not only to the change of workload, but also to the change of virtual machine (VM) configurations. Similar reinforcement learning strategies are used for virtual machine (VM) auto-configuration by VCONF [24]. It automates VM configuration and dynamically reallocates resources allocated to VMS in response to the change of application demands or resources supply.

For session-based Internet services, the slowdown metric is compelling because it characterizes the relative queueing delay, and, it is independent of the session length. Slowdown oriented
resource management was studied on single tier servers. Harchol-Balter [10] designed novel size-based scheduling algorithms that assigned high priorities to requests with small service time so as to improve the performance of servers in terms of mean slowdown. Zhou et al. [42] found a closed form of request slowdown in a $M/G/1$ queueing model and used it for service differentiation on Internet servers. Those slowdown-oriented studies significantly changed the understandings of system performance studies. However, there is no analytical model for session slowdown on multi-tier servers due to inter-tier dependences and per-tier concurrency limits.

In this paper, we propose to use statistical regression models to effectively capture the dynamic behavior of a multi-tier Internet service and to design a novel dynamic resource provisioning strategy that utilizes learned regression models to efficiently provide session slowdown guarantee.

3. Statistical regression analysis

Regression analysis approaches provide a framework to gain knowledge of a computer system, construct system models and make predictions based on the models. It identifies the dependent and independent variables and plots sample values observed for these variables to identify a general data trend without necessarily matching the individual data points. The general trend determines the specific regression analysis such as linear and exponential to be performed on its mathematical representation. Regression analysis results in a quantitative model representation of the relation between the two sets of variables. The quality of the regression model is quantified by statistical measures. One popular technique is to use the correlation coefficient of the model to quantify the “goodness” of the observed data fit to the model. Correlation coefficient is a statistical measure of the interdependence of two or more random variables and its values vary between $-1$ and $+1$. A correlation coefficient of $+1$ reflects a perfect fit with a positive slope between variables, $-1$ reflects a perfect fit with a negative slope and $0$ indicates that the variables are independent of each other. Regression analysis models with qualified fitness can further be used to predict the dependent variable value from one or more measured independent variables.

Using regression based profiling is not new. We use it as a modeling tool for multi-tier Internet service management. Inter-tier dependences, per tier replication and caching constraints, bottleneck tier shift are challenges to multi-tier Internet services. They are further complicated by highly dynamic Internet workloads. It is extremely difficult, if not impossible, to derive a concrete model of a multi-tier system that can effectively capture the complete system dynamics. Regression analysis as a modeling tool provides an attractive alternative solution. Instead of capturing the complete system dynamics, regression analysis captures the patterns and trends of a multi-tier Internet service behavior as simple quantitative models. Thus, regression models can effectively capture Internet service parameter relationships that reflect its behavior when subjected to dynamic workloads.

Importantly, we use regression analysis for predicting resource requirements of a multi-tier Internet service to achieve performance guarantees. Regression analysis is used to capture the session slowdown behavior patterns with respect to the Internet service capacity when subjected to dynamic resource demands. We use the statistical regression model of the “allocated resources–session slowdown” relationship to make accurate resource allocation predictions based on observed value of session slowdown to provide session slowdown guarantees. Similarly, we use regression analysis to quantitatively capture the “allocated resources–resource utilization” relation as a regression model that can be used for resource removal predictions to assure efficient resource utilization in a multi-tier Internet service. We propose to use two distinct statistical regression models to control the upper and lower bounds of the resource allocation.

4. Regression based resource provisioning

We use a statistical regression approach that combines extensive offline training and online monitoring of a multi-tier Internet service metrics to design a new dynamic provisioning strategy for session slowdown guarantee. Offline training is conducted to learn and model the multi-tier Internet service behavior dynamics when subjected to dynamic workloads. During the training phase, regression analysis is used to quantitatively model the Internet service parameter behavior patterns. The statistical regression models along with the workload characteristics are encapsulated by a behavior model. An extensive set of behavior models is learned during the training phase using a diverse set of workloads. The learned behavior models collectively represent the multi-tier Internet service behavior when subjected to dynamic workloads.

When multi-tier Internet service is subjected to real-time workloads, the session slowdown values and resource utilization metrics are monitored and compared with predefined thresholds at periodic time intervals. If a threshold violation is observed, the statistical regression models of a behavior model are used to predict the resource allocation and resource removal requirements of the Internet service based on the observed values of session slowdown and resource utilization. Resources are allocated to or removed from different tiers of the multi-tier service by taking the dynamic resource demands at the individual tiers into consideration.

In the following, we first discuss two metrics, session slowdown and resource utilization. Then we present the structure of a behavior model that encapsulates the Internet service behavior patterns. Next, we discuss the training phase that learns an extensive set of behavior models by conducting statistical regression analysis of Internet service parameter relationships. Finally, we present a novel regression based dynamic resource provisioning strategy for session slowdown guarantee, which makes accurate resource requirement predictions utilizing the learned behavior models.

4.1. Session slowdown

A typical e-commerce application consists of three tiers; a front-end Web tier that is responsible for HTTP request processing, a middle application tier that implements core application functionality say based on Java Enterprise platform, and a backend database that stores product catalogs and user orders. Fig. 1 illustrates the architecture of a three-tier e-commerce server cluster.

A session is a sequence of individual requests of different types made by a customer during a single visit to a web site. In this context, an incoming user request undergoes HTTP processing, application server processing, and triggers queries or transactions at the database. In an $n$-tier architecture, let $d_i$ and $p_j$ denote the queueing delay and processing time of a request $j$ of a session at tier $i$, respectively. The session slowdown is defined as the relative ratio of the total queueing delay of the session requests to the total processing time of the session requests. The rationale is that the session-oriented performance metric should describe the perceived performance at the session level, not at the individual request level. As work in [14,15,31,32], this work assumes that one request at one tier of the multi-tier system does not spawn more than one request to the downstream tier. Thus, a requests total queueing delay is the sum of its queueing delay at the individual tiers and a requests total processing time is the sum of its processing time at the individual tiers. The session slowdown ($s$) is calculated as

\[
    s = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}}
\]

where $m$ is the number of requests in the session.
Table 1
Request compositions in TPC-W.

<table>
<thead>
<tr>
<th></th>
<th>Browsing mix</th>
<th>Shopping mix</th>
<th>Ordering mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browsing request</td>
<td>95%</td>
<td>80%</td>
<td>50%</td>
</tr>
<tr>
<td>Ordering request</td>
<td>5%</td>
<td>20%</td>
<td>50%</td>
</tr>
</tbody>
</table>

Table 2
The shifting bottleneck tier problem.

<table>
<thead>
<tr>
<th>Intervals</th>
<th>Bottleneck tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 4–8, 10–40, 42–50</td>
<td>None</td>
</tr>
<tr>
<td>2</td>
<td>All</td>
</tr>
<tr>
<td>3, 9</td>
<td>Web</td>
</tr>
<tr>
<td>41</td>
<td>Database</td>
</tr>
</tbody>
</table>

4.2. Tier session slowdown ratio

To demonstrate the dynamic behavior of multi-tier Internet applications, we simulate the activities of an e-commerce application using the TPC-W benchmark. Table 1 shows the TPC-W workload that is composed of three distinct session mixes, i.e., Browsing, Shopping and Ordering. Each mix is characterized by different probability based session navigational patterns. Sessions belonging to different mixes visit the tiers in varying number of times with different workload characteristics.

Fig. 2 depicts the utilizations measured at different sampling intervals when the multi-tier service is subjected to a combination of equal number of browsing, shopping and ordering sessions. A server is considered to be the bottleneck server if its utilization exceeds a pre-configured threshold. As shown by Table 2, a different tier becomes the bottleneck tier at certain intervals. For example, during the sampling interval 9, the web tier is the bottleneck. While during the interval 41, the database tier is the bottleneck. The experimental results demonstrate the dynamics challenge for effective per-tier resource allocation for session slowdown guarantee on a multi-tier architecture.

To capture the different resource demands at individual tiers, we define a tier session slowdown as the ratio of the total queuing delay of the requests of the session at a tier to the total processing time of the requests at that tier. That is

\[ S_i = \frac{\sum_{j=1}^{m} d_{ij}}{\sum_{j=1}^{m} p_{ij}}. \]  

A tier session slowdown is affected by the dynamic resource demand on the tier and the resources allocated to the tier. Note that according to the definitions in Eqs. (1) and (2), the tier session slowdowns at individual tiers do not add up to the session slowdown at the multi-tier service level.

One may argue that session slowdown should be modeled at the tier level to represent the resource demand variations. However, modeling the session slowdown at the tier level is not practical because of the inter-tier dependences of the multi-tier architecture. The session slowdown at an individual tier is dependent on the resources allocated at that tier, but also on the resources allocated at the preceding or succeeding tiers. Those dependences are dynamic in nature. Even if the session slowdown can be modeled at the tier level, it can only be used to provide guarantee of tier-level session slowdown instead of user-perceived multi-tier session slowdown.

The normalized tier session slowdown at a tier \( i \) is calculated as

\[ sr_i = \frac{S_i}{S}. \]  

We define the tier session slowdown ratio of an \( n \)-tier service as the ratio of the normalized tier session slowdowns at the individual tiers. That is

\[ sr_1 : sr_2 : \cdots : sr_n. \]  

While a tier session slowdown reflects the resource demand at a tier, the ratio of the normalized tier session slowdowns reflects weighted proportional resource demands on the individual tiers of a multi-tier service. We utilize the ratio to distribute provisioned resources to the individual tiers.

4.3. Resource utilization

We use the term “resource” as an abstract representation of a computing entity with fixed capacity that processes session-based workloads. For example, a virtual machine or a server assigned to the multi-tier Internet service is considered a resource [17]. In this paper, we use the term “resource” to indicate a “virtual server” and use the two terms interchangeably. As others in [4,14,32], we assume that the resources are homogeneous with same capacity and can be assigned to any tier. Resource utilization is the percentage of the resource capacity that is utilized to serve the sessions.
that at least one virtual server is needed at each tier. The load is applied to the multi-tier Internet service multiple times as of TPC-W browsing sessions arriving at 20 sessions/s. The work-
therelations, in the following we consider a workload consisting of a unique behavior model. To demonstrate the statistical regression analysis performed to model the relations, in the following we consider a workload consisting of TPC-W browsing sessions arriving at 20 sessions/s. The workload is applied to the multi-tier Internet service multiple times as the number of allocated virtual servers varying from 3 to 100. Note that at least one virtual server is needed at each tier.

4.4. A behavior model

A behavior model represents the learned behavior of the multi-tier Internet service when subjected to a specific workload. Two important Internet service parameter relations are “allocated resources–session slowdown” and “allocated resources–resource utilization”. A behavior model captures the two parameter relations as quantitative statistical regression modes. With the regression models, the correlation coefficients that quantify the quality of the models are included in the behavior model.

The session mix, session arrival rate and workload characteristics impact the quantitative values of the statistical regression models. Different workloads result in different quantitative representations of the “allocated resources–session slowdown” and “allocated resources–resource utilization” relations. The workload characteristics are integral to a behavior model that captures these relations and is used to distinguish one behavior model from another. A behavior model also captures the tier session slowdown ratio, which indicates weighted proportional resource demands of the workload on the individual tiers.

4.5. The training phase

The training phase is used to observe and quantitatively capture the multi-tier Internet service behavior. An extensive set of behavior models is derived using diverse workloads. We generate workloads for the training purpose according to the industry standard TPC-W benchmark specification. The workload characteristics and the resource variations used during the training phase are summarized in Table 3. Each possible combination of the session type, session arrival rate and allocated resources result in a unique behavior model.

We use statistical regression analysis to derive regression model representations of the “allocated resources–session slowdown” and “allocated resources–resource utilization” relations. To demonstrate the statistical regression analysis performed to model the relations, in the following we consider a workload consisting of TPC-W browsing sessions arriving at 20 sessions/s. The workload is applied to the multi-tier Internet service multiple times as the number of allocated virtual servers varying from 3 to 100. Note that at least one virtual server is needed at each tier.

<table>
<thead>
<tr>
<th>TPC-W session mix</th>
<th>Session arrival rate (per second)</th>
<th>Allocated resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browsing</td>
<td>5, 10, 15, …, 100</td>
<td>3, 4, 5, …, 100</td>
</tr>
<tr>
<td>Shopping</td>
<td>5, 10, 15, …, 100</td>
<td>3, 4, 5, …, 100</td>
</tr>
<tr>
<td>Ordering</td>
<td>5, 10, 15, …, 100</td>
<td>3, 4, 5, …, 100</td>
</tr>
</tbody>
</table>

Fig. 3. “Allocated resources–session slowdown” and “Allocated resources–resource utilization”.

4.5.1. Regression model of “allocated resources–session slowdown”

Fig. 3(a) depicts the session slowdown behavior with the number of virtual servers allocated to the multi-tier system. Because there is no significant change in the session slowdown when the number of allocated virtual servers is greater than 25, those results are omitted in Fig. 3(a). The results reveal a negative exponential growth trend. The negative exponential growth relationship is expressed by Eq. (5), where the variables x and y correspond to the number of virtual servers allocated to the multi-tier Internet service and the observed values of the session slowdown respectively.

\[ y = a_1 e^{-b_1 x} \]  

As the relationship trend is observed to be exponential in nature, statistical exponential regression analysis is performed on Eq. (5). Exponential regression analysis involves linearizing an exponential equation and performing linear regression analysis of the linearized equation. Eq. (5) is linearized by taking its natural logarithm. It yields

\[ \ln y = \ln a_1 - b_1 x \ln e. \]  

Performing linear regression analysis on Eq. (6) results in the following expressions for the coefficients \( a_1 \) and \( b_1 \).

\[ \ln a_1 = \frac{\sum \ln y_i + b_1 \sum x_i}{n}, \]  

\[ b_1 = \frac{n \sum x_i \ln y_i - \sum x_i \sum \ln y_i}{n \sum x_i^2 - (\sum x_i)^2}, \]  

where \((x_i, y_i)\) are the individual data points and \(n\) is the total number of data points plotted in Fig. 3(a).

The numerical values of \( a_1 \) and \( b_1 \) substituted in the Eq. (5) results in

\[ y = 65.483 e^{-0.067 x}. \]  

It represents the quantitative exponential regression model of the “allocated resources–session slowdown” relation.

The quality of the regression model is quantified by the correlation coefficient \( r \). The correlation coefficient for the linearized Eq. (6) using the least square error analysis is

\[ r = \frac{\sum(x_i - \bar{x})(\log y_i - \bar{\log y})}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(\log y_i - \bar{\log y})^2}}. \]  

\((x, y)\) are the individual data points plotted in Fig. 3(a). \(\bar{x}\) and \(\bar{\log y}\) represent the mean of \(x\) and \(\log y\) respectively.

The calculated correlation coefficient \( r \) for the data plotted in Fig. 3(a) is 0.9838. It indicates that the negative exponential
relation is a high quality fit for the observed session slowdown data.

Fig. 3(a) also depicts a visual representation of the data compliance to the regression model. It shows the observed session slowdown values relative to the negative exponential curve. The session slowdown values fit the curve very closely, with most data points being on or very close to the curve.

4.5.2. Regression model of “resources allocated–resource utilization”

Fig. 3(b) depicts the resource utilization behavior with the number of virtual servers allocated to the multi-tier system. As there is no significant change in the resource utilization when the number of allocated virtual servers is greater than 25, those results are omitted in the figure. The results reveal a negative exponential growth relation between the two parameters. The relation is similar to the “resources allocated–session slowdown” relation, but with quantitative differences. The negative exponential growth relationship is expressed by Eq. (11), where the variables x and y correspond to the number of virtual servers allocated to the multi-tier Internet service and the observed values of the resource utilization respectively.

\[ y = a_2 e^{-b_2 x} \]  

Eq. (11) is linearized by taking its natural logarithm. It yields

\[ \ln y = \ln a_2 - b_2 x \ln e \]  

Performing linear regression analysis on Eq. (12) results in the following expressions for the coefficients a_2 and b_2:

\[ \ln a_2 = \frac{\sum \ln y_i + b_2 \sum x_i}{n} \]  

\[ b_2 = \frac{n \sum x_i \ln y_i - \sum x_i \sum \ln y_i}{n \sum x_i^2 - (\sum x_i)^2} \]

where \((x_i, y_i)\) are the individual data points and n is the total number of data points plotted in Fig. 3(b).

The numerical values of a_2 and b_2 substituted in the Eq. (11) lead to a quantitative exponential regression model of the “allocated resources–resource utilization”. That is,

\[ y = 76.2381 e^{-0.0898 x} \]  

The correlation coefficient of the model calculated by applying the data plotted in Fig. 3(b) to the Eq. (10) results in 0.9458, indicating a very good quality fit. The data fit to the negative exponential curve is also presented in Fig. 3(b), which shows the observed resource utilization values relative to the negative exponential curve.

4.5.3. Tier session slowdown ratio

The tier session slowdown ratio reflects the proportional resource demands posed by the workload on the individual tiers. To determine the tier session slowdown ratio, queueing delay and processing time of the individual requests at each tier are first measured. The request level measurements are then used to calculate the multi-tier service level session slowdown, tier session slowdowns, normalized tier session slowdowns using the Eqs. (1)–(3) respectively. Finally the tier session slowdown ratio is calculated using Eq. (4).

The behavior model learned from conducting the training with the specific workload is summarized in the Table 4. Training is repeated with the workloads summarized in Table 3. For each workload instance two regression models are learned. The correlation coefficients of the resulting “resources–session slowdown” regression models range from 0.7951 to 0.9982 with a mean value of 0.87. For the “resources–resource utilization” regression models learned, the correlation coefficients range from 0.7328 to 0.9546 with a mean of 0.82.

<table>
<thead>
<tr>
<th>Table 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A behavior model.</td>
</tr>
<tr>
<td>Session arrival rate</td>
</tr>
<tr>
<td>Session type</td>
</tr>
<tr>
<td>“Resources–session slowdown” regression model</td>
</tr>
<tr>
<td>“Resources–session slowdown” correlation coefficient</td>
</tr>
<tr>
<td>“Resources–resource utilization” regression model</td>
</tr>
<tr>
<td>“Resources–resource utilization” correlation coefficient</td>
</tr>
<tr>
<td>Tier session slowdown ratio</td>
</tr>
</tbody>
</table>

4.6. Dynamic resource provisioning strategy

Regression-based dynamic resource provisioning strategy aims to effectively meet session slowdown guarantee of a multi-tier Internet service under dynamic workloads while ensuring efficient resource utilization. Knowledge of the behavioral dynamics of the multi-tier Internet service is critical for effective and efficient resource management for session slowdown guarantee. The provisioning strategy gains the knowledge from an extensive set of behavior models captured during the training phase.

The resource provisioning process is divided into a sequence of intervals. In each interval, average session slowdown and resource utilization are measured and compared to predefined thresholds. When a threshold violation is observed, resource requirements of the service are predicted using a single learned behavior model. The workload characteristics observed in the interval determine the behavior model used for predictions. A behavior model with the session type same as the dominant session type and session arrival rate closest to the observed session arrival rate is selected. If there are more than one behavior models that meet the criteria, the model with the higher correlation coefficients is selected. The selected behavior model represents the session slowdown and resource utilization behaviors as regression models.

We develop a threshold-based policy that uses a session slowdown threshold and a resource utilization threshold for efficient resource allocation. A session slowdown threshold is set below the session slowdown bound. A threshold violation indicates a possible risk of session slowdown guarantee violation. The “allocated resources–session slowdown” exponential regression model of the selected behavior model is used to predict additional resources required to keep the session slowdown under the threshold in the subsequent intervals. The additional resources are allocated to the individual tiers in proportion to the tier session slowdown ratio of the behavior model.

When there is a resource utilization threshold violation, the “allocated resources–resource utilization” exponential regression model of the behavior model is used to predict the number of virtual servers to be removed. The virtual servers are removed from the individual tiers in inverse proportion to the tier slowdown ratio of the behavior model. Fewer virtual servers will therefore be removed from a tier with relative higher resource demand.

The provisioning strategy is given by Algorithm 1. Table 5 summarizes key notations used.
Algorithm 1 Regression-based dynamic resource provisioning strategy: description

repeat
\[ s_{thr}^{\text{violation}} \leftarrow (s_{\text{avg}} \geq s_{thr}) \] ? true : false
\[ u_{thr}^{\text{violation}} \leftarrow (u_{avg} \leq u_{thr}) \] ? true : false
if \((s_{thr}^{\text{violation}} \lor u_{thr}^{\text{violation}})\) then
  Select a representative behavior model from the set of learned behavior models.
  if \((s_{thr}^{\text{violation}})\) then
    Predict resources to add using “allocated resources–session slowdown” regression model.
    Divide resources among multiple tiers in proportion to the tier session slowdown ratio.
    Allocate additional resources to the various tiers.
  else if \((u_{thr}^{\text{violation}})\) then
    Predict resources to remove using “allocated resources–resource utilization” regression model.
    Divide resources among multiple tiers in inverse proportion to the tier session slowdown ratio.
    Remove resources from the various tiers.
end if
end if
until (ALL SESSIONS PROCESSED)

5. Performance evaluation

5.1. Experimental setup

For performance evaluation, we build a simulation model of a three-tier e-commerce site. It consists of a workload generator consisting of a session generator and a request generator, web servers, application servers and database servers. We evaluate the proposed regression-based dynamic resource provisioning strategy using workloads different than the training workloads. Offline training is conducted using a set of workloads, where each training workload consists of a single traffic mix arriving at constant session arrival rate. For evaluation, we use workloads consisting of combination of different traffic mixes with dynamically varying session arrival rates.

As our previous work on session-based admission control [21] and service differentiation on servers [43], the session-based workload is generated following the guidelines provided by the TPC-W benchmark specification. To generate dynamic workloads with varying session arrival rates, the workload generator takes as input the session arrival rate for each interval. It then determines the number of sessions to launch within the interval. A TPC-W workload session may belong to one of the three distinct traffic mixes, browsing, shopping and ordering. Each of the workload mixes is characterized by different probability based navigational patterns. A session is created as a sequence of interactions for the same customer. For each session of a specific mix, the next interaction is determined by a state transition matrix that specifies the probability of moving from one interaction to another. The session time for the session and think time between the interactions are generated by an exponential distribution with a given mean [5].

The utilization and the processing time of each interaction is derived from the WIRT (Web Interaction Response Time) based on the observation that different mixes pose varying load on the tiers. As others in [2,25], we assume that browsing mix is database tier intensive, shopping mix is application tier intensive and ordering mix is web tier intensive. For example, for the requests of the browsing sessions, the database tier is the most intensive and the web tier is the least intensive. The resource demand of a request on the web tier, application tier and database tier is set as 20%, 30% and 50% of the overall resource demand corresponding to its WIRT. Table 6 summarizes three workloads used for the experiment.

Table 6

<table>
<thead>
<tr>
<th>Workload</th>
<th>Session mix type</th>
<th>Demand intensive tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload-B</td>
<td>TPC-W browsing</td>
<td>Database</td>
</tr>
<tr>
<td>Workload-S</td>
<td>TPC-W shopping</td>
<td>Application</td>
</tr>
<tr>
<td>Workload-O</td>
<td>TPC-W ordering</td>
<td>Web</td>
</tr>
</tbody>
</table>

Fig. 4. A step-change dynamic workload.

Fig. 5. Session slowdown due to regression based dynamic provisioning.

5.2. Session slowdown guarantee

The first experiment is to show that the regression-based dynamic resource provisioning approach effectively provisions resources to meet the session slowdown guarantees of the multi-tier application, but also allocates the resources to the appropriate individual tiers efficiently.

We use three different workload models in Table 6 to examine the performance of the regression based dynamic resource provisioning strategy. Fig. 4 shows a step-change dynamic workload. It is highly dynamic as the session arrival rate increases from 10 to 50 sessions/sec. The total number of virtual servers available for the multi-tier application is 45. The session slowdown bound is set to 5. The session slowdown threshold is set to 3.5. The rationale for selecting the session slowdown threshold value is provided in Section 5.4.

Fig. 5 shows the observed average session slowdown values for the three workload models. Results show that the proposed approach is effective in achieving the session slowdown guarantees of all three workload models for the majority of time. When the session arrival rate reaches 50 sessions/sec, higher values of session slowdown are observed and violations start to happen. We note that by this time all available virtual servers have been allocated to the multi-tier application. When the overload occurs,
Fig. 6. Resource allocation at the overall Internet service and at the individual tiers.

(a) Overall virtual server allocation. (b) Per-tier server allocations: Workload-B. (c) Per-tier server allocations: Workload-S. (d) Per-tier server allocations: Workload-O.

5.3. Efficiency in per-tier resource allocation

We demonstrate that the dynamic provisioning strategy effectively meets the session slowdown guarantees while ensuring efficient resource utilization by the use of a resource utilization threshold.

Fig. 7(a) shows the session arrival rate of the workload used in the experiment. It is highly dynamic as the workload is a random combination of sessions from the three workload models. The session arrival rate dynamically varies from 10 to 40 sessions/sec dynamically. The total number of virtual servers available for the multi-tier application is 45. The session slowdown bound is set to 5. The session slowdown threshold is set to 3.5.

We use the same workload trace to execute the provisioning strategy two times, without and with using the resource utilization threshold. Fig. 7(b) and (c) shows the session slowdown values observed in the two scenarios. In both scenarios, there are very few session slowdown guarantee violations. However, the resource utilization efficiency is very different.

Fig. 8(a) shows the number of virtual servers allocated to the multi-tier application. When no resource utilization threshold is used, the virtual servers once allocated to the multi-tier application are not removed when there is a decrease in the session arrival rate. Next, a resource utilization threshold (70%) is used. In this scenario, the virtual servers are allocated and removed dynamically from the multi-tier application according to the variations in the session arrival rates. Fig. 8(b) shows the number of resources allocated to the multi-tier application. Fig. 8(c) compares the resource utilization in the two scenarios. Apparently, using a threshold achieves much better resource utilization. The experiment demonstrates the threshold-based resource provisioning strategy is capable of achieving session slowdown guarantee while efficiently using the allocated resources.

5.4. Impact of a session slowdown threshold on performance

The session slowdown threshold can significantly affect the performance of the dynamic resource provisioning strategy. A threshold set far below the session slowdown bound may lead
to more threshold violations. This results in more resources provisioned than needed to meet the session slowdown guarantee. On the other hand, if the threshold is set too close to the bound, when threshold violations occur it may be too late to avoid session slowdown guarantee violations by provisioning additional resources.

We conduct an experiment to study the affect of the session slowdown threshold on session slowdown guarantee and on resource utilization efficiency. We change the session arrival rate from 10 to 80 sessions/s. The workload is dynamic and consists of random combination of sessions from the three workload models. The total number of virtual servers available for the multi-tier application is 100. The session slowdown bound is set to 5. We repeatedly execute the experiment using the same workload as the session slowdown threshold changes to at 10%, 20%, . . . , 90% of the session slowdown bound.

Fig. 9(a) shows the number of session slowdown guarantee violations. Fig. 9(b) shows the number of virtual servers allocated to the multi-tier application. The results show that there is no session slowdown guarantee violation when the threshold is below 70% of the bound. However, more virtual servers are allocated to the multi-tier application. As the threshold increases, fewer virtual servers are required to maintain the session slowdown guarantee. However, session slowdown violations start to happen when the threshold is set to 70% of the bound, and, more violations quickly build up as the threshold further increases. It is a trade-off between the resource utilization efficiency and performance guarantee. It deserves a further study on adaptive threshold tuning by statistical learning and control in our future work.

5.5. Impact of a resource utilization threshold on performance

The efficiency of the regression-based dynamic provisioning approach is affected by the resource utilization threshold setup. A low threshold may result in infrequent threshold violations and fewer resource removals, leading to inefficient utilization of the allocated virtual servers. On the other hand, a high resource utilization threshold may lead to premature removal of the virtual servers allocated that may be reallocated in subsequent intervals.

Frequent resource provisioning oscillations, which are successive virtual server removals followed by reallocations, are undesirable. They negatively affect the service performance as each virtual server allocation and removal is associated with a cost in terms of time and processing overhead.

We examine the effect of the resource utilization threshold on the total number of virtual server allocations and removals. The workload used is same as the workload used for analyzing the impact of session slowdown threshold in Section 5.4. We repeatedly execute the experiment using the same workload as the resource utilization threshold is varied at 10%, 20%, . . . , 90%.

Fig. 10 shows the total number of virtual server allocations and removals as the resource utilization threshold is varied. The results show that when the threshold is below 70%, the total number of virtual server allocations and removals remain fairly low. As the threshold is increased above 70% this number increases drastically, indicating that the allocated virtual servers are removed too frequently and are reallocated in the subsequent intervals.

We also observe the session slowdown guarantees violations for specific session slowdown thresholds. We note that the violations observed are similar to those plotted in Fig. 9(a). Using a combination of session slowdown threshold and resource utilization threshold, the regression-based dynamic resource provisioning strategy can effectively achieve session slowdown guarantee while minimizing the resource provisioning oscillations.

5.6. Impact of the online monitoring interval

Each interval of an interval-based provisioning process entails overhead of comparing measured performance metrics to thresholds and predicting resource requirements in case of threshold
violations. The length of the provisioning interval is a trade off between the overhead and the resource requirement prediction accuracy. Short intervals require frequent predictions that lead to increased overheads. Longer intervals may lead to delayed response to the threshold violations, increasing the risk of session slowdown guarantee violations.

To study the effect of the interval length, we use several workload traces with different session arrival rates and consisting of random combination of three different session types. The interval lengths are set to 30, 60 and 90 seconds, respectively. Session slowdown bound is set to 5 and session slowdown threshold is 3.5. The total number of virtual servers available for the multi-tier application is 100.

Fig. 11 shows the observed session slowdown for each possible combination of the workload session arrival rate and interval lengths. Results show that the session slowdown guarantee is met for all workloads when the interval length is set to 30 or 60 seconds. However, the 30-second interval length will lead to more overhead. When the interval length is increased to 90 seconds, violations occur for workloads with higher session arrival rates. However, threshold violations are observed and new resource allocation is triggered only at the end of an interval.

5.7. Validation of regression models and comparison with an analytical model

5.7.1. Validating the statistical regression relations

The training phase described in Section 4.5 utilized TPC-W workloads to derive the “allocated resources–session slowdown” and “allocated resources–resource utilization” statistical regression models. We examine the validity of the derived statistical regression models using an alternative representation of the multi-tier Internet service.

We consider a queueing-based analytical model of a multi-tier Internet service proposed in [31], which represents an n-tier application as a network of n queues processing session based workloads. It proposes a Mean-Value Analysis (MVA) algorithm for closed-queueing networks to compute the response time experienced by a request in a network of queues. The algorithm takes as inputs, the average request service time at each tier $\bar{S}_n$, average think time of a session $\bar{Z}$, and the number of concurrent sessions $N$. It calculates the average queue delay of requests at each tier $\bar{R}_n$, average response time of request $\bar{R}$ and throughput $\tau$ as each session is introduced to the queueing network.

We extend and tailor the algorithm to compute the session slowdown and resource utilization. In accordance with Eq. (1), the session slowdown is calculated as the relative ratio of the total queueing delay of the requests of the session to the total service time of those requests. That is,

$$\frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \bar{R}_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} \bar{S}_{ij}}$$

where $m$ is the number of requests in the session.

The resource utilization is calculated according to the utilization law for a queueing system, which states that $S = \rho / \tau$, where $S$, $\rho$ and $\tau$ are the service time, queue utilization and throughput respectively. That is, resource utilization is calculated as

$$\rho = \tau \sum_{i=1}^{n} \bar{S}_i.$$
We apply the extended MVA-based algorithm to a typical three-tier Internet service, where each tier contains multiple queues that represent the virtual servers allocated to that tier. For different values of the input parameters, we computed the two outputs session slowdown and resource utilization using the extended MVA-based algorithm. For the interest of space and clarity of the figures, we plot the outputs only for two input parameters sets detailed in the Table 7.

For both the input parameter sets, the per-tier request service times are arbitrary as the MVA algorithm does not make any assumption about the service time distributions and the proposed queueing model is sufficiently general to handle workloads with an arbitrary service time requirements [31]. As the work in [31], the user think time of a session is chosen using an exponential distribution and the mean is chosen uniformly at random from the set {1, 5 s}. For the two input parameter sets, we choose the two extreme values, 1 and 5 s as the average user session think times. Choosing sessions with two widely different think times ensures variability in the workload imposed by individual sessions [31].

Fig. 12 shows the session slowdown and resource utilization computed for the two input parameter sets as the total number of virtual servers (queues) allocated to the multi-tier service are varied. Two plots reveal a negative exponential growth trend exhibited by the session slowdown and the resource utilization metrics with respect to the virtual servers allocated to the service. The plots also show the observed session slowdown values relative to a negative exponential curve. They reveal that while the trend observed is negative exponential, the values due to the use of MVA algorithm do not fit the regression curve very closely.

5.7.2. Impact of concurrent sessions on session slowdown

We apply the extended MVA-based algorithm to the typical three-tier Internet service. The number of allocated virtual servers at each tier is five. The number of concurrent sessions is changed from 25 to 500 at increments of 25. The workload consists of a random mix of TPC-W browsing, shopping and ordering sessions. User think time and tier specific service times are chosen to be uniformly random within the range of two sets of parameters specified in Table 7.

We measure the average session slowdown for all completed sessions. Fig. 13 shows the average session slowdown with the number of concurrent sessions. The results show that the session slowdown consistently increases with increase in the workload until the system is saturated at 375 concurrent sessions. We note that at this saturation point, the fixed capacity of the multi-tier Internet service results in sessions being aborted due to the overload.

5.7.3. Comparison with MVA-based dynamic capacity provisioning

We compare the performance of the regression-based resource provisioning approach with the extended MVA algorithm for session slowdown guarantee in the three-tier architecture. The workload used for the experiment consists of random combination of three session types with highly dynamic session arrival rates. The session arrival rate is varied every 30 seconds as shown in Fig. 14(a). Session slowdown bound is set to 5. For the regression based approach, session slowdown threshold is 3.5 and resource utilization threshold is set at 70%.

The MVA algorithm is typically used for steady state workloads. We divide the MVA-based dynamic capacity provisioning process into periodic intervals. The interval length is set to 60 seconds. Within each interval, the MVA algorithm treats the workload as steady state. This was practiced by Urgaonkar et al. [31] to conduct MVA-based dynamic capacity planning for multi-tier Internet services. Similarly, in our approach, at the end of each interval, MVA algorithm is applied to determine the number of servers needed at each tier to satisfy the session slowdown guarantees. The MVA algorithm takes the number of simultaneous sessions to be served and the session slowdown target as the inputs. At each interval edge, the number of simultaneous sessions (N) to be

![Fig. 12. “Allocated virtual servers–session slowdown” and “Allocated virtual servers–resource utilization” relations for a queueing model of a multi-tier Internet service.](image-url)
served in next interval is derived from the peak session arrival rate ($\lambda$) and average session duration ($d$) in the current interval using the Little’s Law ($N = \lambda \times d$).

Fig. 14(b) and (c) show the session slowdown values achieved with the regression and MVA based approaches, respectively. We can observe that the regression-based approach is much more robust to workload variations when providing session slowdown guarantee. A couple of spikes in the session slowdown are observed due to the sudden changes in the applied workload. However, the regression-based approach is responsive in assuring session slowdown guarantee, resulting in much fewer violations compared to the results due to the MVA-based approach.

Fig. 14(d) shows the standard deviation of resulted session slowdown values and the total number of violations for the two resource provisioning approaches. The regression-based approach achieves session slowdown guarantee with lower standard deviation and fewer number of violations than the MVA-based approach does. We acknowledge the robustness to regression-based prediction and use of the session slowdown threshold.

Fig. 14(e) and (f) show the number of virtual server allocated for the multi-tier system with the regression and MVA based approaches respectively. Results show that the regression based approach performs timely virtual server allocation and removal in accordance to the workload changes. The use of the resource utilization threshold results in more efficient resource utilization. Note that the MVA based approach does not consider the removal of the virtual servers [31].

5.8. Impact of TPC-W workload burstiness on performance

Finally we evaluate the effectiveness of the regression based dynamic server provisioning for session slowdown guarantee under a bursty workload. An interesting recent study in [20] discussed a model of bursty e-commerce workloads. It uses a two-state Markovian arrival processes (MAP) to inject burstiness into the TPC-W benchmark. A single parameter, index of dispersion ($I$), is used to control the degree of burstiness. The index of dispersion is used to dynamically modify the think times of a user between submission of consecutive requests within one session.

Using the algorithm proposed in [20] with the dispersion index $I$ set to 300, we generate bursty TPC-W workloads consisting of random combination of three traffic mixes. Fig. 15(a) shows the generated bursty workload. For this experiment, the session slowdown bound is set to 5, the session slowdown threshold is set to 3.5 and the resource utilization threshold is set to 70%. The initial number of virtual servers used is 6, with 2 virtual servers allocated per tier. The maximum number of virtual servers available is set to 60. The online monitoring interval length is set to 20 seconds.

Fig. 15(b) shows the observed session slowdown values. During the first interval, due to the workload spikes the session slowdown guarantee is violated. In the subsequent intervals, the regression based approach effectively predicts the number of virtual servers required to handle the workload changes. The subsequent session slowdown values observed are under the session slowdown bound. However, after about 900 seconds, the system observes higher values of session slowdown. Starting at the 1000th second, session slowdown violation occurs. We note that at the 1000th second, all available virtual servers have been allocated to the system and further capacity increase of the system is not feasible. When overload occurs, we note that our session-based admission control for multi-tier applications designed in [21] can be applied.

Fig. 15(c) shows the number of virtual servers allocated in each interval to process the bursty workload. Note that during the last few intervals, there are many residual requests belonging to the sessions entering in the previous intervals. Thus, more virtual servers are allocated. While the session slowdown guarantee is satisfied after the first interval, bursty workloads result in frequent provisioning oscillations. A provisioning oscillation is allocation of virtual servers in an interval followed by removal of virtual servers in the next interval or vice versa. Frequent resource allocations add overhead to resource management and switching delay to session requests. We plan to explore a coordinated admission control and dynamic provisioning in our future work.

6. Conclusion

Session slowdown is a compelling performance metric of session-based Internet services because it directly measures
user-perceived relative performance. In this paper, we proposed a statistical regression-based approach for effective resource management of a multi-tier Internet service for session slowdown guarantee. We used statistical regression analysis to learn the session slowdown behavior with respect to the Internet service resources. We designed a novel regression-based dynamic resource provisioning strategy that utilizes learned models to predict and manage the resource requirements of the Internet service. Extensive simulation results using TPC-W benchmark workloads have demonstrated the superior performance of the new resource provisioning approach. The regression-based approach adaptively and efficiently provisions resources to appropriate individual tiers for session slowdown guarantee of the multi-tier service, taking the dynamic resource demand and resource utilization into account.

This paper is on session slowdown guarantee with TPC-W workloads in multi-tier Internet services. In our future work, we plan to explore an adaptive threshold strategy using statistical learning to tune the threshold and interval values in the regression-based resource provisioning approach. The generic regression training process will also be expanded with other sophisticated machine learning techniques for performance assurance. We plan to implement the extended approach in a testbed of virtualized server clusters and will report our findings in future related work.

Acknowledgments

This research was mainly supported by US NSF CAREER Award CNS-0844983 and research grant CNS-0720524. Liqiang Zhang’s work was partly supported by NSF grant CNS-0834230. Guihai Chen’s work was partly supported by China NSF grants 60825205, 61073152 and 61133006. A preliminary version of this paper appeared in [22]. The authors thank the anonymous reviewers for their valuable suggestions for revising the manuscript.

References
