Ph.D Thesis Proposal

Multi-tier Internet Service Management: Statistical Learning Approaches

May 2011
Sireesha Muppala
Respectfully submitted by:
Sireesha Muppala
Ph.D. Candidate
3132 Greenmoor Ct
Colorado Springs, CO 80920
smuppala@uccs.edu
sireesha.muppala@oracle.com
mobile 719-640-3274
home 719-282-0277

Date:
May 6 2011

Advisor:
Dr. Xiaobo Zhou

Committee Members:
Dr. Xiaobo Zhou (chair)
Dr. Edward Chow
Dr. Chaun Yue
Dr. Rory Lewis
Dr. Liqiang Zhang
Abstract

Modern Internet services employ a multi-tier architecture. While facilitating flexible service deployment, the tiered architectures introduce several significant challenges. Complex inter-tier dependencies and dynamic bottleneck tier shift are challenges inherent to multi-tier architectures. Highly dynamics of session-oriented Internet workloads further magnifies the complexity of multi-tier Internet services. Managing these complex services for session-oriented performance improvement and quality of service provisioning is an important but challenging undertaking. Three critical Internet service management mechanisms are: admission control, dynamic resource provisioning and service differentiation. Significant research literature studies the three mechanisms using a variety of techniques. Queuing based analytical models of the multi-tier systems have been proposed to achieve request-oriented performance guarantees. However, queuing models often lack the ability to capture session workload dynamics in multi-tier systems and queueing-theoretic approaches are often not effective in resource management for quality-of-service provisioning in highly complex multi-tier systems with highly complex session-based workloads.

This thesis concentrates on applying statistical learning based approaches for multi-tier Internet service management to achieve efficient, balanced and scalable services. The main advantage of statistical learning approaches is that they solve complex dynamic problems through learning and adaptation, and require no priori domain-specific knowledge. First, we will explore a session based admission control strategy to improve session throughput in multi-tier Internet services. Using a bayesian network model, the approach will achieve coordination among multiple tiers of the service for effective session throughput improvement. Second, we will promote the use of a novel relative session-oriented performance metric, session slowdown. Session slowdown represents the user-perceived performance of a session. We will further propose a statistical regression based dynamic resource provisioning strategy for session slowdown guarantee in multi-tier systems. While offline training captures the dynamic behavior of the service as statistical regression models, online monitoring utilizes the learned models to predict the multi-tier Internet service’s resource requirements. Third, we want to design a service differentiation approach that employs a combination of reinforcement learning and neural networks to achieve session slowdown based differentiation in multi-tier Internet services. The approach aims to adaptively achieve both absolute and relative differentiation among multiple customer classes. Finally, we will develop a user interface based Multi-tier Internet Service Management Console, which can be used by a system administrator to monitor and finely tune the service performance. Successful completion of this research will guide using statistical learning approaches for quality-of-service provisioning in modern multi-tier Internet services.
Contents

1 Introduction .................................................... 8
  1.1 The Internet services ........................................ 8
    1.1.1 The multi-tier Internet service architecture .......... 8
    1.1.2 Internet service management ............................ 10
    1.1.3 Challenges ............................................. 11
  1.2 Research focus ............................................ 15
  1.3 Why statistical learning for multi-tier Internet service management .... 18
  1.4 Expected research contributions ........................... 19

2 Related Work ................................................. 19
  2.1 Multi-tier system models .................................. 20
  2.2 Admission control for Internet services ................... 20
  2.3 Dynamic resource provisioning of Internet services .......... 21
  2.4 Service differentiation in Internet services ................. 22
  2.5 Statistical learning techniques for Internet service management .... 23
  2.6 Control theoretic approaches for Internet service management .... 24

3 Proposed Thesis Work ......................................... 25
  3.1 Session based admission control for improved session throughput in multi-tier Internet services ......................... 25
    3.1.1 Bayesian network models ................................ 25
    3.1.2 Bayesian network model of a multi-tier Internet service .... 27
    3.1.3 Coordinated session based admission control (CoSAC) .... 28
  3.2 Session slowdown: A session oriented relative performance metric ...... 29
  3.3 Regression based dynamic resource provisioning for session slowdown guarantees in multi-tier Internet services .......... 30
    3.3.1 Statistical regression analysis .......................... 31
    3.3.2 Regression based dynamic resource provisioning .......... 31
  3.4 Session slowdown based service differentiation in multi-tier Internet services ........................................... 35
    3.4.1 Reinforcement learning & neural networks ................ 36
    3.4.2 System architecture ..................................... 37
    3.4.3 Reinforcement learning for multi-class session service differentiation .... 38
    3.4.4 Reinforcement learning for single-class session slowdown guarantee .... 40
  3.5 Multi-tier Internet Service Management Console ................ 40
    3.5.1 Web application ........................................ 41
4 Experimental Setup 42
4.1 A 3-tier e-commerce service simulator ........................................... 43
4.2 TPC-W workloads ................................................................. 43
4.3 Virtual clustered multi-tier Internet service: A testbed ................. 44
4.4 Software packages .............................................................. 45
4.4.1 Bayesian network modeling ............................................... 45

5 Preliminary Results 45
5.1 Performance evaluation: Session based admission control algorithms .... 45
5.1.1 A blackbox approach ......................................................... 45
5.1.2 Measurement based admission control(MBAC) ...................... 46
5.1.3 Impact of CoSAC on session throughput .............................. 47
5.2 Performance evaluation: Regression based dynamic resource provisioning .... 49
5.2.1 Capturing the multi-tier Internet service behavior: Statistical regression based training ......................................................... 49
5.2.2 Session slowdown guarantees .............................................. 52
5.2.3 Efficiency in per-tier resource allocation .............................. 54

6 Research Plan 55

7 Publications 56
7.1 Conference proceedings ....................................................... 56
7.2 Journal articles ................................................................. 56
7.3 Submitted ................................................................. 56

8 Summary 57
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Multi-tier Internet service architectures.</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>A typical 3-tier Internet service.</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Integrated dependency model for multi-tier Web systems</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Dynamics of a 3-tier e-commerce service.</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Traffic volume intensity variations at an online travel agency.</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>1998 FIFA World Cup trace over ten consecutive days: Burstiness of traffic arrival</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>A multi-tier Internet service with admission control.</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>Dynamic resource provisioning in a multi-tier Internet service.</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>Service differentiation in a multi-tier Internet service.</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>CoSAC: Coordinated session based admission control</td>
<td>26</td>
</tr>
<tr>
<td>11</td>
<td>Bayesian network model of a multi-tier Internet service.</td>
<td>27</td>
</tr>
<tr>
<td>12</td>
<td>An n-tier Internet service.</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>Regression based dynamic resource provisioning</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>Reinforcement learning and neural networks</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>Virtualized datacenter hosting an n-tier Internet service for M customer classes</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>A 3-tier e-commerce simulator.</td>
<td>43</td>
</tr>
<tr>
<td>17</td>
<td>Impact of admission control strategies on session throughput (scenario one)</td>
<td>47</td>
</tr>
<tr>
<td>18</td>
<td>Impact of admission control strategies on session throughput (scenario two)</td>
<td>48</td>
</tr>
<tr>
<td>19</td>
<td>Impact of admission control strategies on session throughput (scenario three)</td>
<td>48</td>
</tr>
<tr>
<td>20</td>
<td>“allocated resources - session slowdown” and “allocated resources - resource utilization” relations.</td>
<td>49</td>
</tr>
<tr>
<td>21</td>
<td>Session slowdown due to regression based dynamic provisioning.</td>
<td>53</td>
</tr>
<tr>
<td>22</td>
<td>Resources allocation at the overall Internet service and at the individual tiers.</td>
<td>53</td>
</tr>
<tr>
<td>23</td>
<td>Impact of using a resource utilization threshold on the session slowdown.</td>
<td>55</td>
</tr>
<tr>
<td>24</td>
<td>Impact of using a resource utilization threshold on the virtual server allocations.</td>
<td>55</td>
</tr>
</tbody>
</table>
### List of Tables

1. The shifting bottleneck tier problem. ................................................. 13
2. CPT for WebTierState node ............................................................... 28
3. Evidences applied to the bayesian network ........................................ 29
4. Regression-based dynamic resource provisioning strategy: notations ....... 35
5. TPC-W transactions ........................................................................ 44
6. Request compositions in TPC-W. ...................................................... 44
7. A behavior model. ........................................................................... 52
8. Experimental workload characteristics. ............................................. 52
1 Introduction

1.1 The Internet services

The Internet has become an indispensable segment of modern day society with billions of users world-wide. The last decade witnessed an explosion of internet users from 250 million in the year 2000 to over 2 billion by the end of 2010 [62]. Individuals, business corporations, academics and government institutions rely heavily on the Internet on a daily basis. Internet services have evolved radically from serving static web pages to delivering highly dynamic and interactive content with extensive multi-media support. Online services such as trading platforms, social networking and instant messaging are now a common occurrence. The influence of e-commerce on national and global economies is undeniable. J.P.Morgan, a leading financial company, forecasts a $235 billion e-commerce revenue for United States for the year 2013 [43]. Ongoing technology innovations, such as cloud computing, are revolutionizing the way small businesses manage their online presence with reduced IT infrastructure costs.

1.1.1 The multi-tier Internet service architecture

Modern Internet services are complex systems that typically employ a multi-tier architecture. In a multi-tier architecture, the functionalities of an Internet service are distributed among two or more tiers, with each tier serving distinct tasks. Each tier provides a certain functionality to its preceding tier and utilizes the functionality provided by its successor [52]. The primary advantage of tiered architectures is ease of service manageability and maintenance. It is possible to repair or upgrade a tier, while the others remain both unaware and unaffected by the change [61]. Other advantages include scalability and reduced data replication.

In a 2-tier architecture, the service functionality is divided between presentation and database tiers. The client program residing in the presentation tier communicates directly with the database server residing in the database tier. The client is responsible for both browser display and application logic execution. The client workstation can thus be optimized for data input and presentation by providing mouse and graphics support. The server can similarly be optimized for data processing and storage with large amounts of disk space and memory. While this simple structure facilitates ease of setup and maintenance, it can cause a bottleneck for data requests. Moreover, the client programs may get too complicated with various application logic rules and result in performance degradation.

The bottleneck and performance problems are addressed by a 3-tier model, where the service functionality is distributed among the web, application and database tiers. In a 3-tier architecture, the complex business logic is moved to the application layer. The result is a less complicated client program residing in the customer’s browser. A single server in the application layer
Figure 1: Multi-tier Internet service architectures.
can handle multiple clients simultaneously, thereby reducing the performance bottleneck effect. Database access is now limited to the application servers where tighter security measures can be employed. Figures 1 (a) and (b) show the tasks handled by each tier and tier interactions in 2-tiered and 3-tiered Internet services respectively.

In multi-tiered Internet services, multiple tiers participate in the processing of each incoming request. Depending on the processing demand, a tier may be replicated using clustering techniques. In such case, a dispatcher is used at each replicated tier to achieve load balancing by distributing requests among the replicas. Figure 2 depicts a typical e-commerce service with three tiers, where the first two tiers are replicated, while the third tier is not. Such an architecture is commonly employed by e-commerce services where a clustered Web server and a clustered Java application server constitute the first two tiers, and the third tier consists of a non-replicable database [52].

More complicated ‘n-tier’ architectures are also possible by separating each tier into multiple functional components. For example, the database tier may be separated into ‘data access’ and ‘data storage’ tiers. In this thesis, we focus on Internet services deployed using a typical 3-tier architecture.

1.1.2 Internet service management

The popularity of multi-tier Internet services continues to grow with an exponentially increasing customer base. Customers demand and expect high performance and quality of service (QoS) guarantees from the Internet services. Most of today’s Internet services provide a service directly to end users. Thus the service success criteria should be defined at the customer level, not just as easily verifiable service-level semantics.

However managing the multi-tier Internet services for optimal performance is non-trivial. The scale and complexity of the Internet services coupled with open-ended Internet workloads result in highly dynamic environments. The service administrators managing the Internet ser-
vices often do not have comprehensive knowledge of the service being managed. Administrators have at best an incomplete view (and at worst a wrong view) of what the services are doing, what they are supposed to be doing and how they are doing it. This leads to unreliable services that are hard to manage [25].

In this thesis, we conduct a systematic study of three critical multi-tier Internet service management mechanisms:

1. Admission control: To regulate the load exercised on Internet service resources by policing and selectively accepting the incoming traffic.

2. Dynamic resource provisioning: To enable real-time adaptation to workload variations through on-demand allocation and removal of resources from an Internet service.

3. Service differentiation: To provide differentiated treatment of multiple customer classes of an Internet service and provide QoS guarantees to one or more high priority classes, irrespective of the service load.

By exploring novel statistical learning based approaches for the three identified management mechanisms, our research strives to achieve improved session-oriented performance and QoS assurances in multi-tier Internet services.

1.1.3 Challenges

While enabling modularity and simplified service deployment, multi-tier architectures also inflict several challenges. We now discuss several challenges encountered in managing multi-tier Internet services.

1. **Inter-tier dependencies**: Multi-tier Internet services exhibit complex inter-tier dependencies, with each tier depending on its preceding tier and providing functionality to its successor. An integrated dependency model that identifies different types of inter-tier software dependencies in a multi-tier system as proposed in [35] is shown in the Figure 3. However, the inter-tier dependency dynamics in their entirety are so complex and complicated that it is even a big challenge to get a good understanding of the entire system dynamic behavior [9, 42].

Moreover, the resource demands posed by user sessions on the individual tiers vary, but are also dependent and correlated to each other. For instance, the web tier may consume mainly CPU and network bandwidth, whereas the database tier consumes more I/O bandwidth than the web tier does. However, a database tier only serves connections established through the web tier [33].
2. **Tier-specific servers and constraints**: Despite being interdependent, each tier of a multi-tier Internet service comprises of distinct servers with distinct characteristics. In a typical e-commerce service, web servers make up the web tier, application tier is composed of Java application servers and database tier hosts SQL servers. While a web server may exclusively handle HTTP protocol, an application server is capable of handling additional protocols like TCP/IP and SMTP. Similarly while the application servers host business objects like EJBs, a database server hosts a database instance like MySQL. Different server types are characterized by distinct performance metrics. Collectively modeling the effects of different type of servers is a complex task [53].

Further complications result from different constraints enforced at different tiers. For instance, not all tiers of a multi-tier Internet service may be replicable. Typically, database tier is difficult to replicate on-the-fly.

3. **Dynamic bottleneck tier shift**: In a multi-tier Internet service, the resource demands posed by user sessions on different tiers is dynamic in nature. Different customers often exhibit different navigational patterns and hence invoke different functions in different ways and with different frequencies[64]. As the client access patterns change, the bottleneck tier dynamically shifts among tiers.

To demonstrate the dynamic behavior of multi-tier Internet services, we simulate the ac-
tivities of an e-commerce service using the industry standard TPC-W benchmark workloads. TPC-W supports three distinct session mixes, Browsing, Shopping and Ordering. Each mix is characterized by different probability based session navigational patterns. Sessions belonging to different mixes visit the tiers varying number of times with different workloads. TPC-W workloads and e-commerce service simulator are further detailed in Section 4.

Figure 4 depicts the tier-specific capacity utilizations measured at different sampling intervals when the multi-tier service is subjected to a combination of equal number of TPC-W browsing, shopping and ordering sessions. A tier is considered to be the bottleneck tier if its capacity utilization exceeds a pre-configured threshold, set to 60% for this experiment. Table 3 captures the observed bottleneck tier for each sampling interval. It is clear that a different tier becomes the bottleneck at different intervals. For instance, during the sampling interval 9 the web tier is the bottleneck, whereas the database tier becomes the bottleneck during the interval 41. This demonstrates the bottleneck shift challenge inherent to multi-tier Internet services.

<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. **Dynamic session length**: Typical customer interaction with an Internet service is captured as a session. A session is a sequence of related requests of different types made by a customer during a single visit to an Internet service. For example, consider a customer’s on-line shopping experience at a retail e-commerce web site. A customer’s session involves multiple requests that search for products of interest, retrieve information about a specific product, add the selected product to the shopping cart, initiate the check-out process, and finally commit the order. In this scenario, performance of the session in its entirety is a critical QoS goal, in contrast to the well adopted individual request-based QoS goals.

Request-based QoS goals typically aim for absolute performance metrics, response time and queueing delay. However, they are not applicable to the session based workloads because of the dynamic session length. Session length, which is the number of requests in a session, is dynamic and is unknown at the time of session origination. Therefore, it is not practical to provide absolute session response time or session delay guarantees. A relative performance metric that is independent of session length is favorable for session based Internet services.

5. **Highly variable workloads**: Internet services experience extreme user demand variations because of the unpredictable nature of the Internet traffic. Predicting the peak workload of an Internet service and capacity provisioning based on worst case estimates is notoriously difficult [54]. Often unforeseeable events such as stock market’s roller coaster ride, terror attacks, Mars landing, can result in a surge of Internet traffic. The unexpected traffic surges can quickly saturate the service capacity and hence affect the
services and even lead to lawsuits due to breakage of service level agreements[64].

Internet workloads exhibit long-term variations such as time-of-day effects as well as short-term fluctuations identified by burstiness (temporal surges). The workload experienced by a top national online travel agency web site is characterized in [40]. The study reveals the traffic patterns and load variations experienced by the web site over a period of 7 days as shown in the Figure 5.

Internet traffic burstiness is explored in [38]. The work analyzes the 1998 FIFA World Cup web-site traces over a period of ten days. The analysis reveals dramatic traffic surges connected to sport events as shown in Figure 6. Burstiness or temporal surges in the incoming requests in an e-commerce server generally turns out to be catastrophic for performance, leading to dramatic server overloading, uncontrolled increase of response times and, in the worst case, service unavailability.

1.2 Research focus

Our primary research goal is effective and efficient management of a multi-tier Internet service for improved session-oriented performance and QoS provisioning. Towards this goal, we explore statistical learning based approaches for three critical management mechanisms: admission control, dynamic resource provisioning and service differentiation, in the context of multi-tier Internet services. We argue for and justify a novel relative performance metric, that is favorable for session-based Internet workloads. Finally, we will develop a user-interface based ‘Multi-tier Internet Service Management Console’, intended to be used by a system administrator to monitor and manage a multi-tier Internet service. We now briefly discuss the individual topics that comprise our research.

When a Internet service experiences transient high user demand, restricting its availability to users is necessary to avoid complete service breakdown. Admission control protects Internet services by preventing resource overload through policing and selective acceptance of incoming traffic. Admission control employed at the request level can certainly protect the service from overload. However it might result in resource wastage in the form of aborted sessions. To
maximize session throughput of a multi-tier Internet service, admission control ought to be employed at session level. A multi-tier Internet service equipped with an admission controller is depicted in Figure 7.

The service quality and responsiveness of a request based system are typically measured by the absolute performance metrics, response time and queueing delay. However, these metrics do not take into consideration the varying demands posed by the different requests. It is known that clients are more likely to anticipate short delays for “small” requests and more willing to tolerate long delays for “large” requests [22]. Request slowdown is the relative ratio of a request’s queueing delay to its service time. 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 [22, 73, 74].

In contrast to the traditionally accepted request oriented performance metrics, we consider the performance of the session in its entirely an important QoS goal. The session length, which is the total number of requests of the session, is unknown at the time of session origination. Due to dynamic session length, it is not practical to guarantee absolute session completion time and session delay of a user session. Instead, a relative performance metric that is independent of session length is appropriate for session based Internet services [74]. We promote a new session-oriented relative performance metric session slowdown, that captures user perceived performance at session level.

Dynamic resource provisioning enables on-the-fly resource management to effectively handle dynamic workloads experienced by an Internet service. Ideally the service should be assigned necessary and sufficient amount of resources to handle its current load [47]. A high level view of dynamic resource provisioning to provide session slowdown assurances in a multi-tier Internet service is depicted in Figure 8. As the observed session oriented performance of the service deteriorates, additional resources are added to satisfy the session slowdown guarantees. Conversely, as the allocated resources become under-utilized they are dynamically removed from the Internet service.
An Internet service can support multiple classes of customers such as premium members, regular subscription members and free users. Traffic from the premium members contributes more to system revenue than traffic from the other classes. Therefore, the priority of serving the sessions initiated by the premium members always precedes that of serving other classes. Service differentiation, a key QoS requirement for multi-tier Internet services [17, 18, 33], is to provide differentiated QoS treatment of multiple customer classes. One way to achieve session slowdown based service differentiation is through biased dynamic resource provisioning to allocate service resources to different customer workloads as shown in Figure 9.

Two approaches to service differentiation are absolute and relative differentiation. In absolute differentiation, a high priority class always receives the desired performance from the Internet service, even in the presence of traffic from low priority classes. This may lead to high priority classes monopolizing the service resources resulting in low priority class starvation. Whereas in relative differentiation, a high priority class is only guaranteed a better performance than a low priority class. While the relative differentiation avoids low-priority class starvation, it can lead to performance degradation with increased demand from a customer class.

Success of an e-commerce business depends on its web site being available and responsive to its customers. Optimizing the performance and availability of such Internet services is a crucial task. Monitoring the service’s resources and viewing the service from an end user’s perspective is critical to ensure that the service is executing with acceptable performance with regards to the customer SLAs. In-depth monitoring can pinpoint existing and potential problems leading to detecting problems before they impact the end users. Ideally, an administrator should be alerted to outages, error conditions and threshold violations before they effect the end users. We plan to develop an user interface based ‘Multi-tier Internet service management console’ that allows an administrator to monitor and manage a multi-tier Internet service in a timely fashion.
1.3 Why statistical learning for multi-tier Internet service management

Statistical learning is concerned with the design and development of algorithms that allow computers to evolve behaviors based on knowledge gained from dynamic observation. The learning strategies come in two flavors: supervised and unsupervised. Supervised learning typically operates in two phases, training and prediction. While training phase is used to gain generalized knowledge about the system under consideration, prediction phase is used to predict the system behavior using the knowledge gained. On the other hand, unsupervised techniques like reinforcement learning are independent of training data and operate by directly interacting with the environment.

The main advantage of applying statistical learning to the multi-tier Internet service management is that the service behavior can be understood through dynamic observation without requiring a priori application-specific knowledge. Secondly, due to the dynamic inter-tier dependencies, dynamic bottleneck tier shift and high variability of the Internet workloads, it is extremely difficult, if not impossible, to derive a concrete analytical model of a multi-tier system that effectively captures the complete system dynamics. Statistical learning as a modeling tool provides an attractive alternative solution, where the behavioral dynamics of the multi-tier service can be “learned” based on observing its operation. Third, dynamic observation of a multi-tier Internet service during the training phase captures its behavior for a large number of distinct workloads. Capturing a large volume of behavior allows us to take advantage of the “law of large numbers” and apply statistical techniques with reasonable confidence [25]. Fi-
nally, by employing statistical learning approaches, a multi-tier Internet service operating in a
dynamic environment gains the ability to learn and adapt to workload variations. Therefore a
system administrator can effectively manage the service without having to foresee and provide
solutions for all possible workload scenarios.

1.4 Expected research contributions

The expected contributions of our research are as follows:

1. A statistical learning based solution for a coordinated session based admission control in
a multi-tier Internet service. The proposed admission control strategy employs a bayesian
network model of the multi-tier Internet service to achieve inter-tier coordination and
improved session throughput.

2. A novel session oriented relative performance metric, session slowdown, which is inde-
dependent of the session length. We promote the use of session slowdown to capture the
user perceived QoS for session based Internet workloads.

3. A statistical regression based dynamic resource provisioning strategy that combines of-
line training with online monitoring for session slowdown assurances in a multi-tier In-
ternet service.

4. A reinforcement learning based approach that adaptively provides absolute and relative
service differentiation for multiple customer classes supported by a multi-tier Internet ser-
vice. We enhance the basic approach with neural network models to facilitate scalability
and agility.

5. A user interface based ‘Multi-tier Internet Service Management Console’, intended to be
used by a system administrator to monitor and manage a multi-tier Internet service.

2 Related Work

Understanding state of the art multi-tier service models and Internet service management mech-
anisms provides an important motivation for our research. There exists an enormous body of
research related to admission control, dynamic provisioning and service differentiation in the
domain of Internet services. In this section we categorize and discuss representative and closely
related research efforts.
2.1 Multi-tier system models

Understanding the entire system dynamics in a multi-tier system is a complex and non-trivial task. Several research efforts attempt to model multi-tier services and target specific QoS goals [17, 66, 44].

A multi-tier system is modeled as a network of queues in [18], that combines layered queuing approaches with a few key approximation-based simplifications. The proposed model is further used to tackle several performance-oriented tasks such as QoS prediction, resource control and capacity planning. A multi-tier web application is abstracted as a an M/GI/1 Processor Sharing queue (M/GI/1/PS) in [31]. It uses a queueing model predictor and enforces admission control of the incoming requests to ensure the desired response time target is met. Diao et.al in [17] propose a Tier-to-Tier (T2T) management architecture that provides a decentralized approach to achieving service level objectives (SLO) in multi-tier systems. The T2T management architecture is structured so that each tier only communicates with its upstream and downstream neighbors. Such an approach avoids the congestion associated with centralized management and improves scalability.

The work in [52] proposed an analytic model for session-based multi-tier services using a network of queues, where the queues represent different tiers of the service. The model can handle services with an arbitrary number of tiers and account for service idiosyncrasies such as replication at tiers, load imbalances across replicas, caching effects, and concurrency limits at each tier. Almeida et.al in [3] use queuing models to represent a virtualization scheme, which partitions physical resources into multiple virtual resources. Each virtual machine is represented by an analytical M/G/1 open queuing model and performance metrics are predicted.

While analytical models exist for multi-tier systems, each model aims for a specific QoS goal, such as bounded response time or request delay. Extending a QoS metric oriented model for another metric, like session slowdown guarantees, is a non-trivial challenge. Moreover, queuing based analytical models often fail to capture the dynamic bottleneck shift and session workload dynamics on multi-tier systems.

2.2 Admission control for Internet services

Admission control for e-commerce services is a non-trivial challenge and receives extensive attention from the research community [8, 10, 12, 30, 76].

Zhou et.al in [67] propose a load shedding mechanism for busy Internet services for overload protection. The work recommends a selective early request termination mechanism to actively detect and abort overdue long requests to improve system throughput. However the focus of this work is a single web server and it aims to improve the request throughput. Our work considers the multi-tiered architecture of the modern Internet services, with the goal of improving session
SBAC is an innovative work on session-based admission control on e-commerce websites [14]. With a simulation model, it shows that an overloaded web server can experience a severe loss of throughput measured as a number of completed sessions compared against the server throughput measured in completed requests. It is able to provide a fair guarantee of session completion, for any accepted session, independent of a session length. However, SBAC was designed for overload control in a single web server and is not effective in a multi-tier server system as the bottleneck tier shifts among tiers when access patterns change dynamically.

The work in [19] deploys an admission controller between the service tier and the bottleneck database tier in a three-tier web site. It identifies different types of servlets and performs overload protection and preferential request scheduling in the form of shortest job first. It assumes that the database tier is the bottleneck and overload control is applied to protect it. However, we argue that assuming a static bottleneck tier is very simplistic and does not represent the true dynamics of a multi-tier system. Effective overload management of a multi-tier system should take into consideration the dynamic bottleneck shift based on the workload experienced.

The work in [20] focuses on session-based admission control for secure dynamic web contents. It recognizes the fact that the cost of establishing a new Secure Socket Layer (SSL) connection is much greater than that of a resumed SSL connection. An admission control approach was designed that prioritizes the resumed SSL connection for performance improvement. Kamra et al. in [24] present Yaksha, a control-theoretic approach for admission control in multi-tiered Web sites that both prevents overload and enforces absolute client response times, while still maintaining high throughput under load. It utilizes a self-tuning proportional integral controller and does not require parameterization of controller weights. The only input required is the desired response time.

The focus of the discussed mechanisms however, is still the response time guarantees at the request level and request based throughput. But improved throughput of requests does not correspond to the improved session throughput. Moreover, the mechanisms discussed do not consider inter-tier coordination in a multi-tier system.

2.3 Dynamic resource provisioning of Internet services

Dynamic resource provisioning for Internet services has been an active research topic in the last decade [3, 9, 17, 23, 24, 37, 50, 52, 54, 57, 60].

Bennani et al. in [7] addresses resource provisioning in large data centers that host several service environments and are subjected to workloads whose intensity varies widely and unpredictably. They use analytic performance models in an efficient manner to design controllers that dynamically switch servers from one service environment to another as needed. In contrast, our
focus is provisioning a single multi-tier Internet service environment taking into consideration the challenges typical to the multi-tier architectures.

The problem of appropriate resource allocation for various service environments in a generic large-scale utility computing infrastructures is addressed by Costa et al. in [16]. They provide a decentralized resource selection service where each compute node is directly responsible for providing accurate and timely information about their resources. To minimize overhead, queries are routed quickly to nodes that can provide the desired resources. Eliminating a centralized resource delegation service represents a simple solution to both implement efficient lookups and support large dimensionality data.

A middleware for controlling performance and availability of cluster-based multi-tier systems, MoKa, is proposed in [4]. It uses an improved analytic model to predict the performance, availability and cost of cluster-based multi-tier services. A utility function based capacity planning algorithm calculates the optimal service configuration, guarantees performance and availability objectives while minimizing functioning cost. The authors of [55] argues that the way the service tier is provisioned can significantly impact a provider’s profit margin. It designs queueing-theoretic methods to provision servers in the service tier with a profit optimization model.

A model-driven server switching policy to dynamically allocate server resources in enterprise systems is proposed in [65]. The multi-tiered architecture is modelled as a multiclass closed queueing network, with each network station corresponding to each service tier. In addition, admission control scheme is used to deal with system overloading, which guarantees that the underlying system can respond to specific customers. Almeida et al. in [2] present a self-managing technique that jointly addresses the resource allocation and admission control optimization problems in virtualized servers. They pose virtualized server management as an optimization problem and provide a solution that takes into account the provider’s revenues, the cost of resource utilization, and customers’ QoS requirements, specified in terms of the response time of individual requests.

2.4 Service differentiation in Internet services

Several research efforts [5, 8, 30, 76, 74, 72, 69, 70, 68, 75] addressed service differentiation in Internet services.

Lee et al. in [29] proposes using traffic shaping and admission control to achieve proportional service differentiation for web servers. Maximum waiting time requirements of multiple clients are effectively met using two distinct admission control algorithms, one client based and the other server based. A web server that can provide differentiated services to clients with different quality of service requirements is considered in [28]. It proposes to using efficient ad-
mission control strategies to achieve effective service differentiation for the clients and to reduce usage costs of the server without violating the QoS requirements.

Queueing based analytical models for request slowdown differentiation on single-tier Internet servers are proposed in [73, 74]. In [73], Zhou et al. derived a closed form expression of the expected request slowdown in an $M/G/1$ FCFS queue with a bounded Pareto service time distribution. [18] proposes a layered queueing model to model differentiated services of multi-tier web services. The service dependencies between multiple tiers, per-tier concurrency limits and resource contention are captured using a $M/M/1$ queueing model.

The problem of quantitative service differentiation on cluster-based delay-sensitive servers is addressed by [71]. The work formulates the problem of quantitative service differentiation as a generalized resource allocation optimization towards the minimization of system delay, defined as the sum of weighted delay of client requests. It derives a closed-form expression of the expected slowdown of a popular heavy-tailed workload model with respect to resource allocation on a server cluster.

The work in [11] formulates a class of constrained optimization problems (QSDL) in the context of a server equipped with multiple levels of service. Solutions to QDSL provide simple rules for dynamically varying the service level to achieve desired trade-offs between output quality and performance. Liu et.al in [33] provide differentiation between multiple multi-tier enterprise services hosted on a shared virtualized datacenter. Request oriented differentiation among multiple services is achieved by employing coordinated resource control at the individual tiers of multiple services. The work assumes of a linear relation between the resource entitlement and performance of a virtual machine.

An extensive literature survey of various admission control, resource provisioning and service differentiation techniques for Internet services is provided in [21].

2.5 Statistical learning techniques for Internet service management

Statistical learning techniques are recently gaining popularity in the domain of Internet services. As multi-tier systems grow in complexity, empirical models built using statistical learning have great potential in overcoming the scalability and complexity challenges [9, 42].

Statistical machine learning techniques have been used to measure the capacity of Internet websites and for online hardware reconfiguration [42, 63]. One approach in [42] uses a bayesian network to correlate low level instrumentation data such as system and user CPU time, available memory size, and I/O status that are collected at run-time to 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.

The work in [13] applies the K-nearest-neighbors (KNN) machine learning approach for
adding database replicas in dynamic content Web server clusters. Experiments using the TPC-W e-commerce benchmark demonstrate that the KNN based proactive scheme is effective in reducing both the frequency and peak level of service-level-agreement violations compared to the traditional reactive schemes.

Statistical learning techniques are also effectively used in configuration and tuning of system parameters. The work in [9] uses 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 strategy is also used for virtual machine auto-configuration by VCONF [41]. It automates the VM configuration and dynamically reallocates the resources allocated to VMs in response to the change of service demands or resources supply.

Tesauro et al. proved the feasibility of statistical learning in autonomic resource allocation [48, 50] and power management [49]. These studies have demonstrated the effectiveness of using statistical learning techniques in system performance detection.

### 2.6 Control theoretic approaches for Internet service management

Control theoretic approaches have been widely applied to various aspects of Internet service management. For instance, a proportional integral controller based admission control proxy was developed in [24] to maintain the average end-to-end delay target. An integration of queuing model with feedback control was applied for average response time control of web systems in [45]. Feedback control was used for service differentiation and performance guarantees on Internet servers [1, 24, 32, 34, 59].

In [27], Lama and Zhou proposed an efficient server provisioning scheme based on an end-to-end resource allocation optimization model for the average and 90th-percentile request delay guarantees. They also designed a model-independent self-tuning fuzzy controller to address the lack of an accurate workload model. The primary goal of the work is to provide request level delay guarantees, in contrast to our goal of session oriented performance guarantees.

Recently, multiple-input-multiple-output (MIMO) control technique has been applied for performance management of multi-tier applications [58] and power control of high density servers in an enclosure [56]. MIMO based approaches can handle the complexity of multi-tier service architecture such as inter-tier dependency, bottleneck switching, as well as the system dynamics of virtualized environments. Padala et al. proposed an automated control of multiple virtualized resources using MIMO control technique [39].

Liu et al. in [33] proposed to use an optimal multivariate control (MIMO) for relative service differentiation between multiple multi-tier applications in a shared hosting platform. The
proposed approach can be tailored for relative service differentiation between multiple service classes in one multi-tier application. However, our service differentiation work differs from it in mainly two aspects. The work in [33] assumes a linear model between the resource entitlement and performance of a virtual machine. It may not be true for all kinds of Internet workloads. Our service differentiation approach makes no such assumptions. Second, the approach in [33] aims to provide request-based service differentiation, while our approach provides both request-based and session-based service differentiation.

3 Proposed Thesis Work

In this section we propose and discuss in detail the various research tasks we plan to execute towards the fulfillment of our research goals.

3.1 Session based admission control for improved session throughput in multi-tier Internet services

When an Internet service is faced with unexpected excessive workloads, admission control is necessary to protect the service resources from overload. It is desirable to employ admission control at session level to minimize aborted sessions and thereby keep the resource wastage to a minimum. However, designing an effective admission control policy for multi-tier Internet services is non-trivial.

User sessions pose varying demands on the individual tiers, leading to some tiers operating under normal load while others are overloaded. For example, studies found that browsing requests tend to put more pressure on the backend database server while ordering related requests exert the least pressure on the database tier [9, 42]. Moreover, the bottleneck tier may shift dynamically as the customer access patterns change. Therefore, admission decisions simply based on the bottleneck tier are ineffective.

An effective admission control mechanism should consider two critical factors: performance of all tiers and coordination among the individual tiers. We propose a novel statistical learning based strategy, Coordinated session based admission control (CoSAC), that takes into consideration both these factors. It utilizes a bayesian network model to achieve coordination among multiple tiers of an Internet service. Figure 10 depicts the framework of the CoSAC strategy.

3.1.1 Bayesian network models

A bayesian network is a high-level representation of a probability distribution over a set of variables that represent a problem domain model. It is a probabilistic graphical model that
Figure 10: CoSAC: Coordinated session based admission control

represents a set of variables as nodes and their conditional probabilistic dependencies as arcs between them, in a parent-child hierarchy. The quantitative relationships between the parent and child nodes are captured by conditional probability tables (CPT). The CPT of each child node captures a collection of probability distributions over the child node, one for each different parental configuration, thus quantifying the parent-child dependency. Bayesian networks offer the following benefits.

- They provide a compact representation of complex problem domains. Since bayesian networks are models of the problem domain probability distribution, they can be used for computing the predictive distribution on the outcomes of possible actions.

- The models have been found to be very robust in the sense that small alterations in the model do not affect the performance of the system dramatically. As such, maintaining and updating existing models is easy since the functioning of the system changes smoothly as the model is being modified.

- Bayesian modeling allows for combing expert knowledge with statistical data in a very practical way. Expert domain knowledge can be coded as prior distributions, that is, the probability distributions can defined independently of processing any sample data. Moreover, all the parameters in bayesian networks have an understandable semantic interpretation. So they can be constructed directly by using domain expert knowledge, without a time-consuming learning process.

- Probabilistic models can handle several different type variables at the same time, whereas many alternative model technologies are designed for some single specific type of vari-
Figure 11: Bayesian network model of a multi-tier Internet service.

3.1.2 Bayesian network model of a multi-tier Internet service

A bayesian network model of a multi-tier Internet service is illustrated in Figure 11. Each oval is a bayesian network node and represents a multi-tier Internet service parameter. The top portion of the node identifies the node by name. The bottom portion lists the mutually exclusive and exhaustive valid node states along with the probability of the node being in each state.

At each tier the utilization and the processing time parameters are captured, leading to a comprehensive representation of the workload states of the tier. This leads to six top level nodes in the model, two per each tier (WebTierUtilization, WebTierProcessingTime and so on). Each of these top-level nodes can be in below threshold (BT), normal range (NR) or above threshold (AT) states. The state is determined programatically based on the comparison of the measured...
runtime parameter value and a configurable threshold value. For instance, if the measured web tier utilization is above the pre-configured threshold, the WebTierUtilization node is in the state AT. Another top-level node is SessionType, the state of which is determined by the type of the incoming session.

For nodes other than the top level nodes, the conditional probability tables play an important role in determining their state. A CPT of a node is defined based on the node inputs and outputs. The node state is determined by processing the applied inputs according to the probabilities defined in its CPTs. Using a simple and efficient process we will train the bayesian network to determine the CPTs of non top level nodes. A sample CPT for the WebTierState node is shown in the Table 2. The first two columns are the inputs to the node, whereas the third, fourth and fifth columns show the probabilities of the node being in the “Normal”, “Underloaded” and “Overloaded” states. The CPT shows that if the WTProcessingTime and WTUtilization node states are NR and BT, respectively, the probability of WebTierState being in either “Normal” or “Underloaded” state is 0.5.

### 3.1.3 Coordinated session based admission control (CoSAC)

We now provide a walk through of the CoSAC strategy using an example. When a new session arrives, the utilization and processing time parameters at each tier and type of the incoming session are recorded. Sample values are shown in Table 3. The measured values are then applied to the appropriate top level nodes as evidence (utilization measured at the web tier is applied as evidence to WebTierUtilization node and so on). The evidence applied to the top-level nodes propagates through the bayesian network through probabilistic inference taking into account the coordinated state of the entire multi-tier system. Based on the sample evidence applied, WebTierUtilization node state is inferred as AT with 100% probability. Similarly, the
Table 3: Evidences applied to the bayesian network.

<table>
<thead>
<tr>
<th>Multi-tier System Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization of the web tier</td>
<td>83%</td>
</tr>
<tr>
<td>Processing time of the web tier</td>
<td>85 ms</td>
</tr>
<tr>
<td>Utilization of the service tier</td>
<td>48%</td>
</tr>
<tr>
<td>Processing time of the service tier</td>
<td>27 ms</td>
</tr>
<tr>
<td>Utilization of the database tier</td>
<td>95%</td>
</tr>
<tr>
<td>Processing time of the database tier</td>
<td>89 ms</td>
</tr>
<tr>
<td>Incoming session type</td>
<td>Shopping</td>
</tr>
</tbody>
</table>

state probabilities of the other nodes are determined by probabilistic inference. According to the probabilistic dependencies defined in the CPTs, the WebTierState and DBTierState nodes are now in Overloaded state with 100% probability. Similarly, the probability of the AppTier-State node being either Underloaded or Normal is 50%. This leads to the state of Shopping for the node SessionsToAccept. The inference process is completed as the state of the AdmitSession node is inferred as Admit with 100% probability. This implies that an incoming shopping session will be accepted and any other type of session is rejected. Note that in this case the probability of admitting the session is 100%. But this may not be the case for other evidence values. As long as the probability of the SessionsToAccept node being in Admit state is above a threshold value (say 60%), the incoming session of will be accepted.

3.2 Session slowdown: A session oriented relative performance metric

Due to dynamic session length, which is unknown at the time of session origination, it is not practical to guarantee absolute performance of a user session in terms of session completion time and session delay. We promote a novel relative metric to represent the QoS of session based workloads. We now define a session-oriented relative performance metric *session slowdown*, that captures user perceived performance at session level.

Consider an alternative representation of an n-tier Internet service illustrated in the Figure 12. 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 \( d_{ij} \) and \( p_{ij} \) 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 requests of the session to the total processing time of the requests. That is,
Figure 12: An n-tier Internet service.

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

where \( m \) is the number of requests in the session, i.e., session length.

Session slowdown is a compelling metric for session-based performance measurement because it is user-perceived service quality at the session level.

3.3 Regression based dynamic resource provisioning for session slowdown guarantees in multi-tier Internet services

Complex inter-tier dependencies and dynamic bottleneck tier shift render the challenge of dynamic resource provisioning for multi-tier systems non-trivial. Blackbox approaches can successfully detect when additional resources need to be provisioned by monitoring QoS violations. However, determining how many resources to provision to which tier is far more complex for multi-tier services [54]. Provisioning oscillations are yet another challenge. Due to rapidly fluctuating loads and slow adaptation of the service, conflicting and unnecessary provisioning operations may be induced. Such oscillations would hurt the overall Internet service performance by incurring time and resource overheads.

Dynamic resource provisioning of a multi-tier Internet service involves two critical and challenging tasks. One, understanding its dynamic behavior when subjected to dynamic workloads and second, adaptive management of its resources to achieve performance guarantees. However, due to the dynamic bottleneck shift and highly variable Internet workloads, it is very difficult to understand the complete multi-tier Internet service behavior dynamics. To address this challenging issue, we propose a statistical learning based approach, statistical regression analysis to be specific. We simplify the complex problem of “how to understand the complete multi-tier
Internet service behavior” to “how to identify and model the dynamic relationships between multi-tier Internet service parameters”.

The goal of the proposed dynamic resource provisioning strategy is to achieve session slowdown guarantees in a multi-tier Internet service operating under dynamic workloads, while ensuring effective resource utilization. Providing session slowdown guarantee on multi-tier servers is quite challenging due to the lack of analytical model for session slowdown on multi-tier servers. The term “resource” is an abstract representation of a computing entity with fixed capacity that can process session-based workloads, for example, a virtual machine. As in [13, 27, 54], we assume that the resources are homogeneous and can be assigned to any tier. Resource utilization is the percentage of the resource capacity that is utilized to serve sessions.

3.3.1 Statistical regression analysis

Statistical regression analysis identifies relationship between dependent and independent variables. Sample values observed for the variables are plotted to identify a general data trend without necessarily matching 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.

We use regression as a modeling tool for dynamic resource provisioning of multi-tier Internet service. Instead of capturing the complex multi-tier dynamics in their entirety, regression analysis captures the patterns and trends of a multi-tier Internet service behavior as simple quantitative models. Regression analysis is used to model the session slowdown and resource utilization behavior patterns with respect to the Internet service capacity when subjected to dynamic resource demands. We consider two service parameter relationships of interest: “allocated resources - session slowdown” and “allocated resources - resource utilization”. We propose to use the two distinct statistical regression models to control the upper and lower bounds of the resources allocated to a multi-tier Internet service.

3.3.2 Regression based dynamic resource provisioning

We propose a regression based dynamic resource provisioning strategy which employs a combination of two phases, offline training and online monitoring. A high-level view of the proposed
strategy is presented in the Figure 13. Each phase is discussed in detail next.

3.3.2.1 Offline training  Training phase is used to observe and quantitatively capture the multi-tier Internet service behavior as it is subjected to dynamic workloads. Offline training is conducted with a diverse set of workloads with each training instance utilizing a specific workload. A single training instance is comprised of the following steps:

- **Regression analysis of “allocated resources - session slowdown” relation**: The multi-tier Internet service is subjected to the training workload repeatedly, as the number of resources allocated to the service vary. With each application of the training workload, the session slowdown value is monitored. Regression analysis is conducted on the collected data pairs (number of allocated resources, session slowdown), resulting in a quantitative relationship $y_1 = f(x)$ where $y_1$ is the observed session slowdown and $x$ is the number of resources allocated to the multi-tier Internet service.

- **Regression analysis of “allocated resources - resource utilization” relation**: In addition to monitoring the session slowdown, with each application of the training workload, resource utilization is also monitored. Regression analysis is conducted on the collected data pairs (number of allocated resources, resource utilization), resulting in a quantitative relationship $y_2 = f(x)$ where $y_2$ is the observed resource utilization and $x$ is the number of resources allocated to the multi-tier Internet service.
Determining the “Tier session slowdown ratio”:

The two relationships modeled in the previous two steps determine the relation between the total number of resources allocated to the multi-tier Internet service and the performance metrics. However, they do not provide any insight into which tier the resources should be allocated to or removed from. 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

$$s_i = \frac{\sum_{j=1}^{m} d_{ij}}{\sum_{j=1}^{m} p_{ij}}.$$  

(2)

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 Eq. (1) and Eq. (2), the tier session slowdowns at individual tiers do not add up to the session slowdown at the multi-tier service level.

It may be argued 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 dependencies 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 dependencies 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

$$sr_i = \frac{s_i}{s}.$$  

(3)

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

$$sr_1 : sr_2 : \cdots : sr_n.$$  

(4)

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.

 Constructing a “Behavior model”:

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

The workload characteristics, session type and session arrival rate affect the quantitative values of the statistical regression models. Different workloads result in different quantitative representations of the parameter relations. The workload characteristics are integral to a behavior model that captures these relations and are used to distinguish one behavior model from another.

Training is repeated with a diverse set of workloads, resulting in an extensive set of behavior models. The set of learned behavior models collectively represent the dynamic behavior of the multi-tier Internet service as it is subjected to dynamic workloads.

### 3.3.2.2 Online monitoring

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 propose 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” 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” 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.
Table 4: Regression-based dynamic resource provisioning strategy: notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s^{thr}$</td>
<td>Session slowdown threshold</td>
</tr>
<tr>
<td>$u^{thr}$</td>
<td>Resource utilization threshold</td>
</tr>
<tr>
<td>$s_{violation}^{thr}$</td>
<td>Session threshold violation (true/false)</td>
</tr>
<tr>
<td>$u_{violation}^{thr}$</td>
<td>Resource utilization threshold violation (true/false)</td>
</tr>
<tr>
<td>$s^{avg}$</td>
<td>Average session slowdown</td>
</tr>
<tr>
<td>$u^{avg}$</td>
<td>Average resource utilization</td>
</tr>
</tbody>
</table>

The online phase of the provisioning strategy is summarized in Algorithm 1 and key notations are provided in Table 4.

Algorithm 1 Regression-based dynamic resource provisioning strategy: description

```plaintext
repeat
  $s_{violation}^{thr} \leftarrow (s^{avg} \geq s^{thr})$ ? true : false
  $u_{violation}^{thr} \leftarrow (u^{avg} \leq u^{thr})$ ? true : false
  if ($s_{violation}^{thr}$ || $u_{violation}^{thr}$) then
    Select a representative behavior model from the set of learned behavior models.
    if ($s_{violation}^{thr}$) 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 ($u_{violation}^{thr}$) 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)
```

3.4 Session slowdown based service differentiation in multi-tier Internet services

Service differentiation is to provide differentiated QoS treatment of multiple customer classes supported by a multi-tier Internet service. Service isolation is to assure that high priority customer class’s performance is not affected by another customer class workload dynamics. Provisioning of service differentiation in a multi-tier Internet service can be achieved by biased allocation of distinct resources to handle workloads from different customer classes. However
it is a non-trivial challenge due to the complexities inherent to multi-tier systems.

We propose a statistical learning based approach that adaptively provides absolute and relative service differentiation in multi-tier Internet services. The proposed approach integrates reinforcement learning and neural networks to learn the real-time resource requirements of multiple customer classes to effectively satisfy the differentiation guarantees. The goal is to provide both request and session oriented differentiation. Integration of neural network models with the reinforcement learning technique allows for improved scalability, efficiency and agility.

### 3.4.1 Reinforcement learning & neural networks

In reinforcement learning (RL), the learner is a decision-making agent that takes actions in an environment and receives reward (or penalty) for its actions in trying to solve a problem. The objective is for the agent to choose actions so as to maximize the expected reward over some period of time. After a set of trial-and-error runs, the agent learns the best policy, which is the sequence of actions that maximize the total reward.

Figure 14(a) illustrates the basic interaction in the normal operation of RL. Each interaction consists of

- observing the system’s current state $s_t \in S$ at time $t$
- performing some legal action $a_t \in A$ in state $s_t$
- receiving a reward $r_{t+1}$, followed by a transition to a new state $s_{t+1}$.

The advantages offered by RL are two-fold. First, RL does not require an explicit model of either the system being managed or the behavioral dynamics of the system. This not only eliminates the need for designing a multi-tier Internet service model, but also eliminates the need to capture complex dynamics of the Internet traffic. Second, due to its basis in Markovian
Decision Process (MDP) theory, it takes into consideration both the immediate and all future rewards of an action. Thus RL could potentially outperform methods that approximate or completely ignore the dynamic effects or cast the decision making problem as a series of unrelated instantaneous optimizations [50].

A neural network consists of a large number of simple processing elements (perceptrons) and exhibits complex global behavior as determined by the weighted connections between the perceptrons. It is a non-linear statistical data modeling tool that can be used to model complex input-output relationships. A multi-layer perceptron (MLP) is a neural network that consists of one input layer, one output layer and one or more hidden layers. A fully connected, feed forward, three layer MLP, with one hidden layer is depicted in Figure 14(b). The MLP shown has ‘n’ inputs and generates ‘m’ outputs, thereby mapping an n-dimensional space into an m-dimensional space.

Application of reinforcement learning techniques is non trivial due to the exponentially increased state space with table-based Q-value learning when systems scale up. An alternative to table based learning is to use a non-linear continuous value approximation. Several design choices are available for the approximation, neural networks, regression, kernel methods to cite a few. Due to the lack of prior knowledge of the value function properties, we opt for neural network as a function approximator [50]. In online system management, interaction-based RL policy suffers from slow adaptation to new policies. It has been shown that model-based RL is more data efficient [6]. Without explicitly storing the training instances, a model can achieve a good approximation and allows generalization. We once again choose an neural network model because of its ability to capture complex non-linear input-output parameter relationships.

3.4.2 System architecture

For the challenge of service differentiation we consider a multi-tier Internet service hosted in a virtualized data center. Server virtualization is often used to ensure high resource utilization in multi-tier applications [26, 36]. Virtualization enables a data center to accommodate multiple Internet applications, as well as to provide performance guarantees for each hosted application [33].

A multi-tier application hosted in a virtualized data center is depicted in the Figure 15. Each virtual machine in this setup is dedicated to a specific tier serving a customer class. For a n-tier application supporting M customer classes, the data center consists of n X M virtual machines. The virtual machines may be co-located on a single physical server or spread across multiple physical servers.
3.4.3 Reinforcement learning for multi-class session service differentiation

We propose to achieve absolute and relative service differentiation of multiple customer classes through biased allocation of dedicated resources to each customer class. The resources considered are the CPU and memory of the virtual machines hosting the multi-tier Internet service. To resolve the dynamic CPU and memory allocation problem using reinforcement learning, we first define the problem in terms of state, action and reward.

State Set

The state set captures the CPU capacity and memory configurations of the $n \times M$ virtual machines hosting the multi-tier service.

$$s = (\text{cpu}_{11}, \text{mem}_{11}, \cdots, \text{cpu}_n, \text{mem}_n, \cdots, \text{cpu}_{M1}, \text{mem}_{M1}, \cdots, \text{cpu}_{Mn}, \text{mem}_{Mn})$$  \hspace{1cm} (5)

where $\text{cpu}_{ij}, \text{mem}_{ij}$ represent CPU and memory of the virtual machine representing $j^{th}$ tier dedicated to $i^{th}$ customer class.

Action Set

For each of the two configurable parameters of a virtual machine, possible operations are either $\text{increase}(+1)$, $\text{decrease}(-1)$ or $\text{nochange}(0)$. All possible combinations of the three actions applied to the two configurable parameters of all virtual machines result in a complete action set. A sample action

$$a = (\text{cpu}_{11}(0), \text{mem}_{11}(+1), \cdots, \text{cpu}_{Mn}(0), \text{mem}_{Mn}(0))$$  \hspace{1cm} (6)
indicates an increase in the memory of the virtual machine hosting the first tier dedicated to
customer class 1.

With each action, only a single configuration parameter of a virtual machine is modified. This
resembles the natural trial-and-error method closely and searches the state space exhaust-
ively. An action is deemed invalid, if it results in the total cpu/memory allocated to the virtual
machines exceeding the available cpu/memory.

**Reward Function**

The reward depends on the learner’s mode of operation. As long as cpu and memory are avail-
able to be allocated to the virtual servers, the learner operates in absolute mode and aims to
provide absolute differentiation guarantees. When the cpu and memory resources are exhausted,
the learner switches to relative mode and aims to provide relative service differentiation to the
multiple customer classes. The reward takes into consideration the observed performance of
customer classes and utilization of the virtual machine resources.

A class utility function relates the observed performance of the class to its guaranteed tar-
gets. In absolute mode, the class utility is defined as

\[ U^A_i = e^{\lambda(s^*_i - s_i)} + e^{\lambda(r^*_i - r_i)} \]  

where \( s^*_i, s_i \) are the desired and observed session slowdowns, \( r^*_i, r_i \) are the desired and
observed mean request response times for class \( i \).

In relative mode, the class utility is defined as

\[ U^R_i = e^{\alpha(s^*_i - s_{i+1}) + r^*_i - r_{i+1}} \]  

where \( s^*_i, s_i \) are observed session slowdowns and \( r^*_i, r_i \) are observed response times
for classes \( i+1 \) and \( i \) respectively. \( \alpha \) is the desired ratio of relative performance and controls the
relative differentiation between two subsequent classes.

From Eqns. 7 and 8 it is clear that utility of a class is large when the observed performance
metrics are less than the performance targets. Conversely, the utility of the class drops when the
observed performance metrics are greater than the performance targets. The exponential utility
function has an inherent momentum feature. As better performance is achieved, the class utility
achieved for a given action will grow quickly, thus promoting better and faster convergence
toward better QoS [51].

We next define an utility function that relates the observed and desired resource utilization
of a virtual machine,

\[ RU_k = e^{c_{uk} - c_{u^*}} + e^{m_{uk} - m_{u^*}} \]  

39
where \( c_{uk} \) and \( m_{uk} \) are observed CPU and memory utilization of the \( k^{th} \) virtual machine and \( c^*, m^* \) are the desired CPU and memory utilization for all virtual machines. Eqn. 9 results in high utility when the virtual machine resources are effectively utilized and in low utility when their utilization drops below the expected targets.

Finally, the reward is defined as cumulative utility of the customer classes and the resource utilization.

In absolute mode the reward is

\[
r = \sum_{i=1}^{M} U^A_i + \sum_{k=1}^{nXM} RU_k
\]

and in the relative mode, the reward is

\[
r = \sum_{i=1}^{M} U^R_i + \sum_{k=1}^{nXM} RU_k
\]

### 3.4.4 Reinforcement learning for single-class session slowdown guarantee

As we pursue a reinforcement learning based solution for session slowdown differentiation guarantees among multiple customer classes through dynamic resource provisioning, we recognize an opportunity to apply the same technique to achieve session slowdown guarantees for a single customer class. This is essentially the challenge tackled by ‘Regression based dynamic resource provisioning for session slowdown assurance’ in section 3.3.

While the statistical regression simplifies the multi-tier Internet service dynamics, it suffers from two disadvantages. First, the learned regression models are tightly coupled with the TPC-W workload characteristics. Second, the learned behavior models are persisted during training phase and queried during the online phase, thus incurring storage and lookup overheads.

As a secondary task of the service differentiation research, we will explore reinforcement learning based dynamic resource provisioning for session slowdown assurances of a single customer class.

### 3.5 Multi-tier Internet Service Management Console

Multi-tier Internet services and the underlying infrastructure can be very complex involving a wide range of servers, networks, databases, operating systems, and third-party web services. Keeping these web-based services running with acceptable performance presents a challenge to the IT teams responsible for them. Real-time performance monitoring is at the heart of ensuring that these systems deliver the results expected from the investment they require.

Developing an exhaustive monitoring and profiling console that conducts real time monitoring of all involved components is a huge undertaking. Given the limited scope and time
constraints of a PhD thesis, it is unreasonable to attempt a full pledge monitoring console as a single research task. Instead, we choose to develop a console that integrates with our previous research and focuses on configuration/monitoring of the admission control and resource provisioning mechanisms in a multi-tier Internet service.

For this final research task, we foresee two important goals. First, we propose to develop a web-based user interface (UI) application, Multi-tier Internet Service Management Console, intended to be used by an administrator. The management console allows an administrator to monitor and police the session traffic and manage resources allocated to various tiers of a multi-tier Internet service. Second, we propose to explore a combined admission control and resource provisioning strategy to improve the stability and performance of a multi-tier Internet service. The two proposed research goals are briefly discussed next.

3.5.1 Web application

The proposed web application consists of five major UI views, each dedicated to a specific feature. The five supported features are admission control management, resource provisioning management, performance monitoring and report generation.

3.5.1.1 Admission control management view Will provide an administrator with a view session traffic statistics, the number of incoming, accepted, completed, rejected and aborted sessions. Using this UI view, the administrator can fine tune several thresholds (e.g., web tier utilization thresholds to determine the state of the WebTierUtilization node) with the goal of maximizing the session throughput.

3.5.1.2 Resource provisioning management view Will provide an administrator with a comprehensive view of resources allocated to various tiers, along with available free resources. The utilization and memory graphs of the resources are available at the individual resource level and at each tier level. The resource provisioning strategy implemented is the reinforcement learning based approach as discussed in section 3.4.4. The administrator can fine tune several thresholds with the goal of meeting the session slowdown guarantees while ensuring efficient resource utilization.

3.5.1.3 Alarm management view Will allow an administrator to configure alarms for one or more interesting events. For example, if a resource is operating at greater than 90% utilization for more than a specific period of time, the administrator desires to be notified. The priority of the alarm (High, Medium, Low) and its delivery mechanism (Email/Pager/View in the management console) will be configurable.
3.5.1.4 **Performance monitoring view**  Will provide a view of real time behavioral trends of performance metrics like session slowdown, resource utilization, throughput. This real time monitoring allows an administrator to measure and tune the performance of the service to ensure that productivity and revenue goals are being met.

3.5.1.5 **Reports view**  Periodic (daily and weekly) reports showing performance trends, outage statistics, and SLA compliance are generated automatically by the management console. An administrator will be able to choose to view the reports within the management Console or receive them via email.

3.5.2 **An integrated admission control and resource provisioning approach**

Admission control and resource provisioning are complementary to each other in two ways. First, while resource provisioning allocates the resources, admission control protects the allocated resources by policing the incoming sessions. Second, while admission control determines the workload experienced by the multi-tier service resource provisioning allocates resources to meet the workload. We propose to explore a combined ‘admission control + dynamic resource provisioning’ strategy in the following two scenarios.

3.5.2.1 **Finite available resources**  As the available free resources of an Internet service are depleted, continued session traffic will result in SLA violations. Admission control combined with resource provisioning is likely to result in fewer SLA violations, by reducing the load on the service resources.

3.5.2.2 **High intensity bursty traffic**  When the Internet service is faced with high intensity traffic bursts, timely resource provisioning to satisfy the SLA guarantees may not be possible. Continued short traffic bursts over a time period are likely to result in provisioning oscillations. Employing a combined admission control and resource provisioning strategy is one way to reduce the oscillations.

4 **Experimental Setup**

For successful completion of our research, we intend to conduct an extensive evaluation of the proposed statistical learning based strategies. In this section we detail the experimental setup used to conduct the evaluation experiments. We plan to execute a combination of simulation and implementation experiments to evaluate the proposed session based admission control, dynamic
resource provisioning and service differentiation strategies. The various components used for evaluation throughout our research are detailed below.

4.1 A 3-tier e-commerce service simulator

We develop a Java based multi-threaded simulator model of a three-tier e-commerce service to evaluate the proposed Internet service management strategies. The simulator consists of a customer generator, a session and a request generator. It follows a multi-tier architecture as in Figure 16 and consists of multiple web servers, service servers and database servers. Without being affected by the methods of implementation, the simulator can effectively evaluate the performance of the proposed strategies.

4.2 TPC-W workloads

The session-based workload processed by the e-commerce simulator is generated according to the guidelines provided by the TPC-W benchmark specification. TPC-W benchmark [46] is an industry standard transactional web benchmark workload. TPC-W defines 14 different transactions. These transactions can be roughly classified as ‘Browsing’ or ‘Ordering’ type, as shown in Table 5. Furthermore, TPC-W defines three standard traffic mixes, Browsing, Shopping and Ordering, based on the weight of each transaction type in the particular traffic mix as shown in Table 6. 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. Typically, a user session starts with a Home transaction request. The session time for the session and think time between the interac-
Table 5: TPC-W transactions

<table>
<thead>
<tr>
<th>Browsing Type</th>
<th>Ordering Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Shopping Cart</td>
</tr>
<tr>
<td>New Products</td>
<td>Customer Registration</td>
</tr>
<tr>
<td>Best Sellers</td>
<td>Buy Request</td>
</tr>
<tr>
<td>Product Detail</td>
<td>Buy Confirm</td>
</tr>
<tr>
<td>Search Request</td>
<td>Order Inquiry</td>
</tr>
<tr>
<td>Execute Search</td>
<td>Order Display</td>
</tr>
<tr>
<td>Admin Request</td>
<td>Admin Request</td>
</tr>
<tr>
<td>Admin Confirm</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Request compositions in TPC-W.

<table>
<thead>
<tr>
<th></th>
<th>Browsing</th>
<th>Shopping</th>
<th>Ordering</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>

...tions are generated by an exponential distribution with a given mean [14].

4.3 Virtual clustered multi-tier Internet service: A testbed

We have a prototype virtualized data center which consists of 12 HP ProLiant BL460C G6 blade server modules and a 40 TB HP EVA storage area network with 10 Gbps Ethernet and 8 Gbps Fibre/iSCSI dual channels. Virtualization of this cluster is enabled by VMWares vSphere 4 Enterprise edition. The operating system on each VM is Ubuntu Linux version 10.04. vSphere controls the disk space, memory, and CPU share (in MHz) allotted to the VMs, and also provides an service programming interface (API) to support the remote management of virtual machines. We want to implement a virtualized multi-tier server cluster as shown in Figure 12 assuming that the database tier is not replicable.

We plan to use an open-source multi-tier service, RUBiS, in our experimental studies. RUBiS implements the core functionality of an eBay like auction site: selling, browsing and bidding. It implements three types of user sessions, has nine database tables and defines 26 interactions that can be accessed from the client’s Web browsers. The service contains a Java-based client that generates a session-oriented workload. RUBiS sessions have an average duration of 15 minutes with an average think time of 5 seconds. It defines two workload mixes: a browsing mix made up of only read-only interactions and a bidding mix that includes 15% read-write interactions.
4.4 Software packages

4.4.1 Bayesian network modeling

Netica software [15] is used to model multi-tier Internet service as a Bayesian network. Netica is a powerful and easy-to-use software for working with belief networks and influence diagrams. It uses the fastest known algorithm for exact general probabilistic inference in a compiled Bayesian network, known as “message passing in a junction tree of cliques”. Netica-J is the Java API that can be used in conjunction with the Netica software. Our simulator integrates with the bayesian network model using the Netica-J libraries.

5 Preliminary Results

Our preliminary evaluation efforts focussed on establishing the feasibility of statistical learning approaches in addressing the challenges of session-based admission control and dynamic resource provisioning in complex multi-tier Internet services. In this section we detail the evaluations conducted and discuss the observed results.

5.1 Performance evaluation: Session based admission control algorithms

The goal of our first set of evaluations is to prove the feasibility of a statistical learning based solution for session-based admission control in multi-tier Internet services. We conduct a simulation based evaluation using the e-commerce simulator discussed in section 4.1. We provide a comparison of the proposed CoSAC strategy with two other admission control strategies, a black box approach and a measurement based approach.

5.1.1 A blackbox approach

We consider a blackbox approach which is a straightforward extension of SBAC, a widely accepted session-based admission control approach [14]. We extend SBAC, originally designed for a single web server, to a multi-tier architecture.

The algorithm is as follows. Various tier utilizations are measured at regular time intervals. Based on the measured utilizations in the recent past intervals, the utilization of each tier for the next interval is predicted using the exponential moving average method. The predicted tier utilizations are then compared with pre-configured tier-specific utilization thresholds. As soon as the predicted utilization of any tier exceeds the threshold, new sessions are rejected until the predicted utilization falls below that threshold in subsequent intervals. Essentially, the admission control mechanism treats the multi-tier service as a blackbox. The admission decision is based on the utilization of the bottleneck tier, whichever it is at decision time.
5.1.2 Measurement based admission control (MBAC)

With the Blackbox approach, in any given interval, either all or none of the new sessions are accepted. This can lead to underutilization of the service resources at certain tiers, while only one of the tiers is overloaded and the other tiers are operating at a normal load. MBAC aims to overcome this limitation by pro-actively accepting different traffic mixes based on the predicted utilizations of the individual tiers. This is motivated by the observation that the requests of different types pose different resource demands on a tier. Studies found that browsing requests tend to put more pressure on the database tier while ordering related requests exert the least pressure on the database tier [42]. As others in [9, 42], we assume that browsing mix is database tier intensive, shopping mix is service tier intensive and ordering mix is web tier intensive. MBAC achieves a more balanced utilization of the tier resources and improves the effective session throughput by accepting a mixture of different sessions.

The algorithm is as follows. Two utilization thresholds per tier are maintained, one for minimum and one for maximum. Various tier utilizations are measured at regular time intervals. Based on the measured utilizations in the recent past intervals, the utilization for each tier for the next interval is predicted using the exponential moving average method. At the interval edge, the admission control decision is made for the next interval based on the predicted utilization of the tiers.

- If the predicted utilizations of all tiers are below their minimum threshold values, all new sessions are accepted.
- If the predicted utilizations of all tiers are above their maximum threshold values, no new sessions are accepted.
- If the predicted utilizations of all tiers are in between the minimum and maximum threshold values, new sessions belonging to different mixes are accepted in proportion to the predicted utilization ratio of the tiers.
- If only one of the tier utilizations is above its maximum threshold, new sessions belonging to different traffic mixes will be accepted in proportion to the ratio of other two predicted tier utilizations.
- If only two of the predicted tier utilizations are above their maximum thresholds, only Browsing, or Shopping, or Ordering sessions are accepted, depending on which tier’s predicted utilization is below the maximum threshold.
5.1.3 Impact of CoSAC on session throughput

We now evaluate the impact of CoSAC with a bayesian network on the session throughput and compare it with the results of Blackbox and MBAC approaches. The workload consists of an equal number of Browsing, Shopping and Ordering TPC-W sessions. We conducted experiments at different session arrival rates between 10 to 100 sessions/sec. Figures 17(a)(b)(d)(f) show the number of accepted, rejected, completed, and aborted sessions, respectively. Results demonstrate that CoSAC is able to accept and complete significantly more sessions than the MBAC and Blackbox approaches. When the overall session arrival rate is greater than 80 sessions/sec, the saturation point of the website is reached. Using the results due to the Blackbox as the baseline, Figure 17(c) shows that CoSAC is able to accept as many as 45% more sessions and MBAC is able to accept about 20% more sessions. Figure 17(f) shows that CoSAC is able to complete as many as 50% more sessions and MBAC is able to complete as many as 22% more sessions. This demonstrate the significance of multi-tier admission control approaches. Particularly, the admission coordination between multiple tiers via the bayesian network representation leads to significant session throughput improvement.

Figure 18 shows the results due to the use of a workload consisting a 3:2:1 ratio of Browsing, Shopping and Ordering sessions. The plots have the basic same shapes as those in Figure 17. But the overall session throughput is slightly lower. We believe this is because with the equal number of sessions of different types, the load is more uniformly distributed to the different tiers.
(a) Accepted sessions.  
(b) Rejected sessions.  
(c) Acceptance improvement.  
(d) Completed sessions.  
(e) Aborted sessions.  
(f) Completion improvement.

Figure 18: Impact of admission control strategies on session throughput (scenario two).

(a) Dynamic session arrivals.  
(b) Accepted sessions.  
(c) Acceptance improvement.  
(d) Aborted sessions.  
(e) Completed sessions.  
(f) Completion improvement.

Figure 19: Impact of admission control strategies on session throughput (scenario three).
of the system. With the sessions of different types in more different ratios, some tiers are more loaded than the other tiers, leading to more rejected and aborted sessions. Nevertheless, the CoSAC outperforms other two approaches significantly with respect to the completed sessions (the effective session throughput).

The experiments above use traffic traces that are generated at static session arrival rate. We also studied the performance of the admission control strategies under a dynamic workload. Figure 19(a) depicts the dynamic workload. The overall session arrival rate changes from 10 sessions/sec during the first 10 seconds to 50 sessions/sec in the last 10 seconds. Figures 19(b)(d)(e) shows the number of accepted, aborted, completed sessions, and 19(c)(f) shows the improvements in percentage.

5.2 Performance evaluation: Regression based dynamic resource provisioning

In this section, we report the results of simulation based training and evaluation of the proposed dynamic resource provisioning strategy.

5.2.1 Capturing the multi-tier Internet service behavior: Statistical regression based training

To demonstrate the training phase of the regression based provisioning, we detail the steps involved as the e-commerce simulator is subjected to a specific workload. The workload used consists of TPC-W browsing sessions arriving at 20 sessions/second. The workload is applied to the multi-tier Internet service multiple times as the number of allocated virtual servers is varied from 3 to 25. At least one virtual server is needed at each tier.
Regression model of “allocated resources - session slowdown”

Figure 20(a) depicts the session slowdown behavior with the number of virtual servers allocated. It reveals a negative exponential growth relationship, which is expressed by Eq. (12). 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.

$$y = a_1 e^{-b_1 x}. \quad (12)$$

As the relationship trend is observed to be exponential in nature, statistical exponential regression analysis is performed on Eq. (12). Exponential regression analysis involves linearizing an exponential equation and performing linear regression analysis of the linearized equation. Eq. (12) is linearized by taking its natural logarithm. It yields

$$\ln y = \ln a_1 - b_1 x \ln e. \quad (13)$$

Performing linear regression analysis on Eq. (13) results in the following expressions for the coefficients $a_1$ and $b_1$,

$$\ln a_1 = \frac{\sum \ln y_i + b_1 \sum x_i}{n}. \quad (14)$$

$$b_1 = \frac{n \sum x_i \ln y_i - \sum x_i \sum \ln y_i}{n \sum x_i^2 - (\sum x_i)^2}. \quad (15)$$

where $(x_i, y_i)$ are the individual data points and $n$ is the total number of data points plotted in Figure 20(a).

The numerical values of $a_1$ and $b_1$ substituted in the Eq. (12) results in

$$y = 65.4833e^{-0.067x}. \quad (16)$$

Eq. (16) is 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. (13) using the least square error analysis is

$$r = \frac{\sum (x_i - \bar{x})(\log y_i - \log \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (\log y_i - \log \bar{y})^2}}. \quad (17)$$

$(x_i, y_i)$ are the individual data points plotted in Figure 20(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 Figure 20(a) is 0.9838. It indicates that the negative exponential relation is a high quality fit for the observed session slowdown data.
The Figure 20(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.

Regression model of “allocated resources - resource utilization”
Figure 20(b) shows the resource utilization behavior as the number of virtual servers allocated to the Internet service change. The plot reveals 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. (18), 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.

$$y = a_2 e^{-b_2 x}.$$  \hspace{1cm} (18)

Eq. (18) is linearized by taking its natural logarithm. It yields

$$\ln y = \ln a_2 - b_2 x \ln e.$$ \hspace{1cm} (19)

Performing linear regression analysis on Eq. (19) results in the following expressions for the coefficients $a_2$ and $b_2$.

$$\ln a_2 = \frac{\sum \ln y_i + b_2 \sum x_i}{n},$$ \hspace{1cm} (20)

$$b_2 = \frac{n \sum x_i \ln y_i - \sum x_i \sum \ln y_i}{n \sum x_i^2 - (\sum x_i)^2},$$ \hspace{1cm} (21)

where $(x_i, y_i)$ are the individual data points and $n$ is the total number of data points plotted in Figure 20(b).

The numerical values of $a_2$ and $b_2$ substituted in the Eq. (18) lead to a quantitative exponential regression model of the “allocated resources - resource utilization”. That is,

$$y = 76.2381 e^{-0.098x}.$$ \hspace{1cm} (22)

The calculated correlation coefficient of the model by applying the data plotted in Figure 20(b) to the Eq. (17) is 0.9458, indicating a very good quality fit. The data fit to the negative exponential curve is also presented in Figure 20(b), which shows the observed resource utilization values relative to the negative exponential curve.

Tier session slowdown ratio
To capture the proportional resource demands on the individual tiers, Tier session slowdown ratio is calculated as discussed in Section 3.3.
Table 7: A behavior model.

<table>
<thead>
<tr>
<th>Session arrival rate</th>
<th>20 sessions/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session type</td>
<td>TPC-W Browsing</td>
</tr>
<tr>
<td>&quot;allocated resources - session slowdown&quot; regression model</td>
<td>$y = 65.4833e^{-0.067x}$</td>
</tr>
<tr>
<td>&quot;allocated resources - session slowdown&quot; correlation coefficient</td>
<td>0.9938</td>
</tr>
<tr>
<td>&quot;allocated resources - resource utilization&quot; regression model</td>
<td>$y = 76.2381e^{-0.0989x}$</td>
</tr>
<tr>
<td>&quot;allocated resources - resource utilization&quot; correlation coefficient</td>
<td>0.9458</td>
</tr>
<tr>
<td>Tier session slowdown ratio</td>
<td>0.25 : 0.16 : 0.59</td>
</tr>
</tbody>
</table>

Table 8: Experimental workload characteristics.

<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>Service</td>
</tr>
<tr>
<td>Workload-O</td>
<td>TPC-W Ordering</td>
<td>Web</td>
</tr>
</tbody>
</table>

Behavior model

A behavior model represents the learned behavior of the multi-tier Internet service when subjected to a specific workload. The behavior model resulted from training with a specific workload is summarized in the Table 7.

5.2.2 Session slowdown guarantees

Our next experiment is to show that our regression based dynamic resource provisioning strategy not only effectively provisions resources to meet the session slowdown guarantees of the multi-tier service, but also allocates the resources to the appropriate individual tiers efficiently.

In the experiment, we use three different workload models in Table 8 to examine the performance of the regression based dynamic resource provisioning strategy. Figure 21(a) shows the dynamic session arrive rate of workloads. It is highly dynamic as the session arrival rate increases from 10 sessions/sec to 50 sessions/sec. The total number of virtual servers available for the multi-tier service is 45. The session slowdown bound is set to 5. The session slowdown threshold is set to 3.5.

Figure 21(b) shows the observed average session slowdown values for the three workload models. Results show that the dynamic resource provisioning strategy is effective in achieving the session slowdown guarantees of all three workload models for the majority of time. When
Figure 21: Session slowdown due to regression based dynamic provisioning.

(a) A dynamic workload.  
(b) Session slowdown.

Figure 22: Resources allocation at the overall Internet service and at the individual tiers.

(a) Overall virtual servers allocation.  
(b) Per-tier server allocations: Workload-B.  
(c) Per-tier server allocations: Workload-S.  
(d) Per-tier server allocations: Workload-O.
the session arrival rate reaches 50 sessions/second, 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 service.

Figure 22(a) shows the number of virtual servers provisioned to the multi-tier service for the three workload models. There are no considerable differences for different workloads. This is due to the fact that in all three cases, workloads of similar session arrival rate are applied to the multi-tier service and the regression based resource provisioning strategy effectively distributed the virtual servers to appropriate tiers as three workload models impose different resource demands on different tiers. Next we examine the resource allocation at the individual tiers for each workload and the differences become apparent.

Figure 22 (b) shows the number of virtual servers allocated at different tiers with Workload-B. It shows that more virtual servers are allocated to the database tier than those allocated to the web and service tiers. This agrees with the observation that the browsing sessions impose more resource demand on the database tier than the other two tiers.

Figure 22(c) shows the number of virtual servers allocated at different tiers with Workload-S. It shows that the majority of virtual servers are allocated to the service tier. Figure 22(d) shows that the major of provisioned resources are allocated to the web tier for Workload-O. The results demonstrate that the trained regression model accurately captures the workload dynamics and is effectively utilized by the dynamic resource allocation provisioning strategy for session slowdown guarantees.

5.2.3 Efficiency in per-tier resource allocation

Next 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.

Figure 23(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 sessions/sec to 40 sessions/sec. The total number of virtual servers available for the multi-tier service is 45. The session slowdown bound is set to 5. The session slowdown threshold is set to 3.5.

Using the same workload trace we execute the provisioning strategy two times, without and with using the resource utilization threshold. Figures 23(b) and 23(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.

Figure 24(a) shows the number of virtual servers allocated to the multi-tier service. When no resource utilization threshold is used, the virtual servers once allocated to the multi-tier
service 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 service according to the variations in the session arrival rates. Figure 24(b) shows the number of resources allocated to the multi-tier service. Figure 24(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 guarantees while efficiently using the allocated resources.

6 Research Plan

As mentioned throughout this document, our research is made up of a systematic study of multi-tier Internet service management with a focus on session based admission control, dynamic resource provisioning, and service differentiation. In each of these focused areas, we will continue to conduct a critical survey of published research literature. We propose novel statistical learning strategies for improved performance and QoS guarantee satisfaction in multi-tier Internet services. Our research deliverables include design, evaluation and publication of the proposed strategies. As discussed in the previous section, a preliminary evaluation of our
session based admission control and statistical regression based provisioning strategies has lead to encouraging results. We plan to conduct the next set of detailed evaluations on a virtualized multi-tier Internet service testbed. We expect the refinement of the proposed strategies and their evaluation to continue through Fall’2011. We will measure the success of the research both by the results achieved and the recognition we receive through conference and journal publications. We will start the development of Multi-tier Internet Service Management Console in Spring 2012. Formal documentation of the proposed strategies and evaluation results is expected to start in Summer 2012 and will be completed no later than Aug 2012. However, throughout the research, as results or findings of interest become available, we will seek to document our results in major international journals and conferences. We intend to defend the research in summer or fall semester in 2012.

7 Publications

Our research to date has been recognized by the research community and lead to the following publications.

7.1 Conference proceedings


7.2 Journal articles


7.3 Submitted

8 Summary

In this document, we present our proposal to apply statistical learning techniques towards the management of dynamic multi-tier Internet services. In the face of challenges typical to multi-tier architectures, it is our goal to propose novel algorithms for session-based admission control, dynamic resource provisioning, service differentiation and service isolation in Internet services. We articulate the need for and justify a novel session-oriented relative performance metric, *session slowdown*, that is favorable for session based Internet workloads. We plan to conduct extensive evaluation of the proposed management strategies to prove the feasibility and effectiveness of statistical learning techniques in the domain of multi-tier Internet services. We are encouraged by the positive results obtained during the preliminary evaluation of proposed session-based admission control and resource provisioning algorithms. We expect our research tasks to continue through Summer 2012 and aim for a Fall 2012 defense.

Acknowledgement

This research was supported in part by U.S. National Science Foundation CAREER Award CNS-0844983 and research grant CNS-0720524.
References


