Coordinated Session-based Admission Control with Statistical Learning for Multi-tier Internet Applications

Sireesha Muppala and Xiaobo Zhou
Department of Computer Science
University of Colorado at Colorado Springs, CO 80918
{sdasaraj, xzhou}@uccs.edu

Abstract

Popular Internet applications deploy a multi-tier architecture, with each tier provisioning a certain functionality to its preceding tier. In this paper, we address the challenging session-based admission control issue for multi-tier Internet applications. The session-based admission control approach (SBAC) designed for a single Web server is not effective for a multi-tier architecture. This is due to the fact that the bottleneck in a multi-tier website dynamically shifts among tiers as client access patterns change. Admission control based on only the bottleneck tier is not efficient as different sessions impose different resource consumptions at the different tiers. First, we propose a multi-tier measurement based admission control (MBAC), which pro-actively accepts different session mixes based on the utilization state of all tiers. More importantly, we design a coordinated session-based admission control approach (CoSAC) based on a machine learning technique. It uses a Bayesian network to correlate the states of all tiers. The probability with which a session is admitted is determined by the probabilistic inference of the network after applying the evidence in terms of utilization and processing time at each tier to the network. We compare CoSAC with MBAC and a black-box approach tailored from SBAC, using the industry standard TPC-W benchmark in a typical three-tier e-commerce website. Experimental results demonstrate the superior performance of CoSAC with respect to the effective session throughput.

Keywords: scalable Internet architecture, session admission control, statistical learning, TPC-W benchmark, effective session throughput

1 Introduction

Due to the dynamic nature and scale of the Internet, Internet applications pose great challenges including scalability and availability [19]. Today, popular Internet applications deploy a multi-tier architecture, with each tier provisioning a certain functionality to its preceding tier and making use of the functionality provided by its successor to carry out its part of the overall request processing [1, 4, 5, 8, 9, 10, 11, 14, 15, 16, 17, 20]. The multi-tier computer systems keep growing in scale and complexity. They become so complicated that it is even a big challenge to get a good understanding of the entire system dynamic behaviors [12, 13].

Admission control is a critical mechanism for peak load management and quality-of-service provisioning for Internet services. Most popular Internet applications are session-based. A session is a sequence of individual requests of different types made by a customer during a single visit to a website. SBAC (session-based admission control) is an important approach on e-commerce websites [2]. It originally proposes to use session throughput, instead of request throughput, to evaluate the web server performance. However, SBAC is not effective for peak load management in a multi-tier architecture. This is mainly due to the fact that the bottleneck tier in a multi-tier website dynamically shifts among tiers as client access pattern changes.

The industry standard TPC-W benchmark has browsing, shopping, and ordering mixes. One may argue that under heavy load conditions, only ordering sessions be accepted since they are more likely than other sessions to result in economic benefits for an e-commerce site. But our experiments found that accepting only ordering sessions under a heavy load condition is not the right way to utilize the resources in a multi-tier website. Processing a request involves multiple system components at the different tiers. Requests of different session mixes impose different resource consumptions at the different tiers. For example, studies found that the browsing related requests tend to put more pressure on the backend database server while the ordering related requests tend to put the least pressure on the database tier [12, 13]. Therefore, designing the admission control simply based on the bottleneck tier is not efficient.

In this paper, we propose two new session-based admission control approaches for multi-tier Internet applications. We first propose a multi-tier measurement based admission control (MBAC), which proportionally accepts different session mixes based on the utilization state of the tiers. The motivation is that the saturation of the system in processing one type of requests may not necessarily mean that the system cannot handle other types of requests. Accepting a mixture of different types of sessions may make the multi-tier servers more balanced. Thus, the effective session throughput defined as the number of completed sessions can be improved.

More importantly, we propose a coordinated session-based admission control approach (CoSAC) with a machine learning technique. As the multi-tier server system becomes more and more complicated, people envision that empirical models built using statistical learning have great potentials to help overcome the challenges of scale and complexity [1, 13, 12, 18]. CoSAC achieves the admission control coordination among all tiers by modeling the multi-tier system as a Bayesian network. The probability with which a session is admitted is determined by the probabilistic inference of the network after applying the evidence in terms of utilization and processing time at each tier to the Bayesian network. This results in coordinated admission decisions, by taking into account the state of all tiers at the time of admission.

We compare CoSAC with MBAC and a black-box approach tailored from SBAC. We evaluate the approaches using the TPC-W workload benchmark in a typical three-tier e-commerce environment. Experimental results demonstrate the superior performance of CoSAC. It can improve the effective session throughput by about 50% compared to the black-box approach in most scenarios, while MBAC can improve that effective session throughput by about 20%.

The paper is organized as follows. Section 2 reviews the related work. Section 3 describes the design of admission control approaches. Section 4 is on the performance evaluation. Section 5 concludes the paper.

2 Related Work

Multi-tier server architectures impose challenges including shifting bottleneck tier, inter-tier dependency, and per-tier concurrence limit. Resource management for multi-tier Internet applications has been a very active research topic [8, 9, 15, 16, 17]. There are studies that focused on the modeling and analysis of multi-tier servers with queueing foundations [5, 4, 15]. Dynamic server provisioning is a useful technique for QoS provisioning and capacity planning in multi-tier servers. For instance, Urgaonkar et al. proposed a novel dynamic server provisioning technique on multi-tier clusters. It applies a queueing model to determine the number of servers to allocate to each tier of the multi-tier application [16]. The work in [17] designed queueing-theoretic methods to provision servers in the application tier with a profit optimization model. Our work focuses on the session-based admission control for multi-tier applications, which complements the above findings.

SBAC is an innovative work on session-based admission control on e-commerce websites [2]. 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 also reveals that the overloaded web server may discriminate against longer sessions, which however are more likely to result in purchases. SBAC is able to provide a fair guarantee of session completion, for any accepted session, independent of a session length. SBAC was designed for overload control in a single web server. It is not effective in a multi-tier server system as the bottleneck tier shifts among tiers when access patterns change dynamically.

The work in [6] deploys an admission controller between the application tier and the bottleneck database tier in a three-tier website. 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. But it does not provide any coordination between multiple tiers for the session admission control. The work in [21] studied a load shedding mechanism for busy Internet services. It deploys a selective early request termination mechanism to actively detect and abort overdue long requests to improve system throughput. The work in [7] 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. The mechanisms discussed above however do not provide any coordination between multiple tiers for session-based admission control.

Statistical machine learning techniques are recently used to measure the capacity of Internet Websites and for online hardware reconfiguration [12, 13, 18]. One approach in [12] 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 [1] 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. Those studies have demonstrated the effectiveness of using machine learning techniques in system performance detection. In this paper, we use a machine learning technique for coordinated admission control on a multi-tier architecture to improve the effective session throughput.

3 Session Admission Control

3.1 Dynamics of a multi-tier website

A typical e-commerce application consists of three tiers; a front-end Web tier that is responsible for HTTP request processing, a middle application tier that implements core application functionality say based on Java Enterprise platform, and a backend database that stores product catalogs and user orders. In this context, an incoming user request undergoes HTTP processing, application server processing, and triggers queries or transactions at the database.

We simulated the activities of a business oriented transactional website using TPC-W workloads to demonstrate that the bottleneck tier of the three-tier website dynamically shifts. Table 1 shows the TPC-W workload that is composed of three distinct session mixes, Browsing, Shopping and Ordering. Each of these mixes is characterized by different probability based session navigational patterns. Sessions belonging to different mixes visit the tiers in varying number of times and pose

Table 1: Request compositions in TPC-W.

	Browsing	Shopping	Ordering
Browsing request	95%	80%	50%
Ordering request	5%	20%	50%

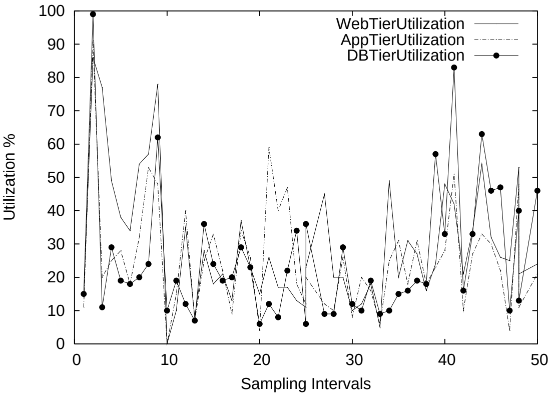


Figure 1: Dynamics of a three-tier website.

different loads, thereby leading to imbalance in the tier utilizations.

Figure 1 depicts the utilizations measured at different sampling intervals. A server is considered to be the bottleneck server if its utilization exceeds a pre-configured threshold. As shown by Table 2, a different tier becomes the bottleneck tier at certain intervals. For example, during the sampling interval 9, the web tier is the bottleneck. while during the interval 41, the database tier is the bottleneck. The experimental results demonstrate the dynamics challenge for effective session-based admission control on a multi-tier architecture.

3.2 A Blackbox Approach

This approach is a straightforward extension of SBAC, a widely accepted session-based admission control approach [2]. SBAC was designed for a single web server. This blackbox approach extends it to a multi-tier website. The utilization of various tiers is constantly monitored at sampling intervals. Using the measured utilizations in the recent past intervals, the predicted utilization for each tier

Table 2: The shifting bottleneck tier.

Intervals	Bottleneck Tier
9	Web
41	Database

for the next interval is calculated. The predicted utilizations are compared with the pre-configured tier-specific utilization thresholds. Based on the comparison results, the admission control decision is made whether to accept or reject new sessions in the next interval. In this approach, as soon as the predicted utilization of one tier exceeds the maximum threshold, it stops accepting new sessions until the predicted utilization falls below that threshold in subsequent intervals. Essentially, the admission control treats the multi-tier website as a blackbox. The admission decision is based on the utilization of the bottleneck tier, whichever it is in a multi-tier website.

3.3 The MBAC Approach

In a multi-tier e-commerce website, processing a request involves multiple system component in different tiers. Saturation of the system in the processing of one type of requests may not necessarily mean it cannot handle other requests. With the Blackbox approach, in any given interval, either all of the new sessions are accepted or none of them are accepted. This can lead to under utilization of the system resources at certain tiers, while only one of the tiers is overloaded and the other tiers are operating at a normal load. We propose a multi-tier measurement based admission control (MBAC) approach, which aims to overcome the limitation by accepting different traffic mixes based on the utilizations of the individual tiers.

The idea is to pro-actively accept different session mixes in proportion to the predicted utilizations at the different tiers. This is motivated by the observation that the requests of a mix type (e.g., Browsing mix) puts more pressure on a specific tier while the requests of another mix type (e.g., Ordering mix) tends to put the least pressure on that tier [12]. Based on the statistical analysis of workload characteristics, MBAC may achieve more balanced utilization of the tier resources and improves the effective session throughput (completed sessions) by accepting a mixture of different sessions. Two utilization thresholds, one for minimum and one for maximum are maintained. Once the minimum utilization of one tier is exceeded, incoming sessions belonging to different traffic mixes are proportionally accepted, to keep the tier utilizations in balance and improve session throughput.

The basic procedure is as follows. At sampling intervals, the utilization of each tier is measured. The predicted utilization of a tier for the next interval is computed based on the measured utilization in a number of previous intervals based on an exponential moving average method. At the end of each interval, 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, any new session will be admitted in the next interval.
- If the predicted utilizations of all tiers are above their maximum threshold values, no new sessions will be admitted in the next interval. Requests belonging to the already admitted sessions will be accepted.

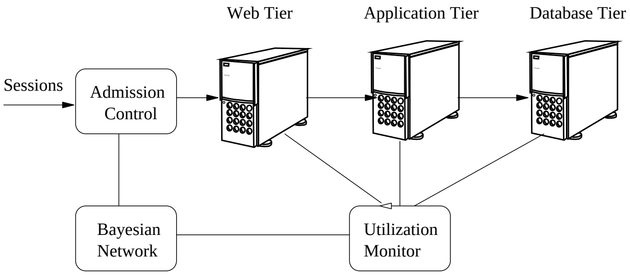


Figure 2: The system architecture.

- If the predicted utilizations of all the tiers are in between the minimum and maximum threshold values, new sessions belonging to different mixes will be accepted in the next interval in proportion to the predicted utilization ratio of the tiers.
- If one of the tier utilizations is above its maximum threshold, in addition to the requests belonging to the already accepted sessions, new sessions belonging to different traffic mixes will be accepted in a proportional manner according to the ratio of other two tier utilizations. For example, assuming that Browsing mix is database tier intensive, Shopping mix is application tier intensive and Ordering mix is web tier intensive. If the predicted utilization of the web tier exceeds the maximum threshold and the predicted utilizations of the application and database tiers are below the maximum thresholds, no sessions from Ordering mix will be accepted but sessions from Browsing and Shopping mixes will be accepted in proportion to the predicted utilization of the database and application tiers.
- If two of the tier utilizations are above their maximum thresholds, the session to admit is either Browsing, or Shopping, or Ordering, depending on which tier utilization is below the threshold.

3.4 The CoSAC approach

The Blackbox and the MBAC approaches suffer from the following issues:

- The admission control decision is completely based on measured values of the utilizations at individual tiers. There is no coordination between the states of the various tiers.
- The admission control decision is executed at the interval edges. At the time of the admission control decision, if it is decided that no new sessions will be accepted, none will be accepted until the decision is changed at end of the interval, even if some tiers are no longer overloaded during the interval.

We propose a coordinated admission control approach (CoSAC) with a machine learning technique to overcome the above shortcomings. Figure 2 illustrates the system architecture. Instead of making the admission control decision at a specific interval, the decision will be made each time a new

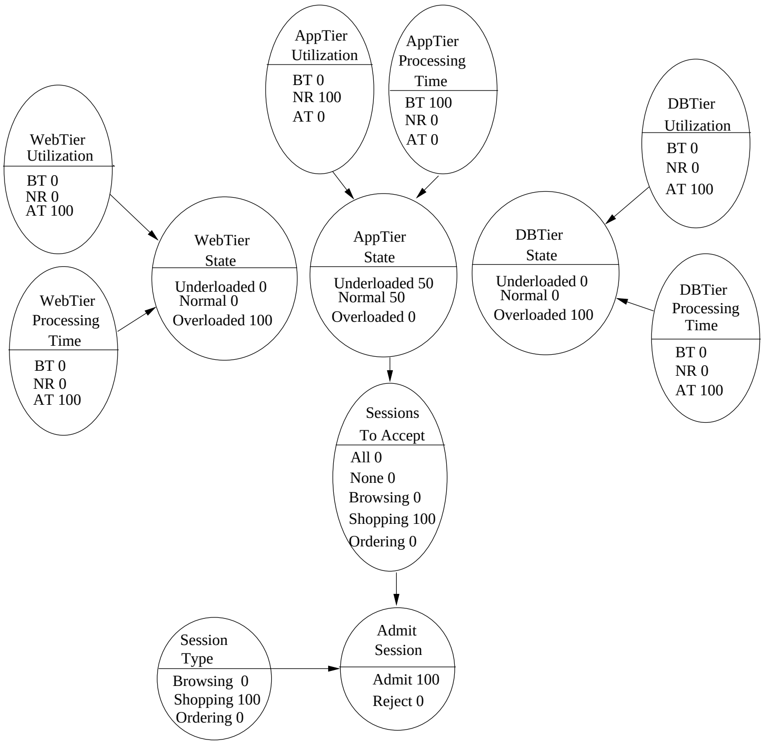


Figure 3: A Bayesian network and an instance.

session enters the system. Coordination between various tiers is achieved by modeling the system as a Bayesian network. The probability with which a session is accepted or rejected is determined by the probabilistic inference of the network after applying the evidence in terms of utilization and processing time at each tier to the network. This leads to a coordinated decision, by taking into account the state of all three tiers at the time of decision.

A Bayesian network is a probabilistic graphical model that 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 means of 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.

The Bayesian network representation of the multi-tier system is illustrated in Figure 3. Each oval represents a parameter of the system, with the top portion specifying the name and the bottom portion specifying mutually exclusive and exhaustive valid states of the Bayesian network node. Each tier has a utilization parameter and a processing time parameter, that lead to a more comprehensive representation of the workload states of the tier. For example, the utilization parameter measured at the web tier is represented by the network node WebTierUtilization. It can be in either Below Threshold (BT), within Normal Range (NR) or Above Threshold (AT) state. The processing time

Table 3: Evidences applied to the Bayesian network.

Multi-tier System Parameter	Value
Utilization of the web tier	83%
Processing time of the web tier	85 ms
Utilization of the application tier	48%
Processing time of the application tier	27 ms
Utilization of the database tier	95%
Processing time of the database tier	89 ms
Incoming session type	Shopping

parameter at the web tier is represented by the network node WebTierProcessingTime. It can also be in either BT, NR or AT states.

The numerical values shown in Figure 3 are the probability of the network node being in the state when a specific evidence in terms of a measured utilization or processing time is applied to the Bayesian network. For instance, when one new session arrives, the utilization and processing time parameters of each tier are measured, which are shown in Table 3. The measured value of the web tier utilization is applied as an evidence for WebTierUtilization node, the measured application tier utilization as an evidence for AppTierUtilization node, and so on for all other parameters. The evidences applied to these top-level nodes are propagated through the Bayesian network. It executes the probabilistic inference, taking into account the coordinated states of the entire multi-tier system. Based on the evidence applied to the top-level network nodes, WebTierUtilization node state is inferred as AT with 100% probability. Similarly, the state of other nodes and its probability will be determined by the probabilistic inference. Because of the probabilistic dependencies defined by the CPTs of the WebTierState and DBTierState nodes, both these nodes now have the Overloaded state with 100% probability. Similarly, the AppTierState node has two states, i.e., Underloaded and Normal each with 50% probability. This leads to the state of Shopping for the node SessionsToAccept. The type of the incoming session is applied as an evidence to the node SessionType, which gives the node the state of Shopping. The inference process is finished as the state of the AdmitSession node is inferred as Admit with 100% probability. This results in the incoming session of shopping type to be accepted. If the incoming session is of any other type, it would be rejected. We note that in this case the probability of admitting the session is 100%. But this may not be the case for other values of the evidence. As long as the probability of the SessionsToAccept node with the state Admit is above a threshold value (say 60%), the incoming session of any type will be accepted.

3.5 Conditional Probability Tables (CPTs) and Bayesian Network Training

Determining the state of a Bayesian Network node is an essential part of the probabilistic inference of a Bayesian Network. CPTs play an important role in this process and are associated with all nodes of the Bayesian Network. We define the CPTs for each node based on its inputs and output. The state of a node is determined by processing the applied inputs according to the probabilities defined in its CPT.

There are six top level nodes in the Bayesian Network, i.e., WebTierUtilization, WebTierProcessingTime, AppTierUtilization, AppTierProcessingTime, DbTierUtilization and DBTierProcessingTime. The state of each node is determined programatically based on the configurable threshold values. Each node can be in BT, NR or AT states. For instance, if the measured web tier utilization is above the threshold, the WebTierUtilization node is in the state AT. The utilization and processing time parameters (such as web tier utilization and web tier processing time) defined at each tier are the measured values in the experiment. As a session's individual requests are processed at each tier, the tier's measured utilization and processing time values are updated based on the resource demands posed by those individual requests.

For nodes other than the top level nodes, we use a simple and efficient process for training the Bayesian Network to determine the CPT tables. The training process involves conducting experiments with several variations of CPTs for different values of the threshold utilizations. The workload used for training consists of equal number of Browsing, Shopping and Ordering sessions. Based on those experiments, CPTs for all nodes except the top level nodes are determined for the performance evaluation of the CoSAC strategy.

For reference, we provide one such CPT determined by the training process in Table 4. The table represents the CPT of the WebTierState node. The first two columns are the inputs to this node and 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 nodes are in NR and BT states respectively, the probability of WebTierState being in either "Normal" or "Underloaded" state is 0.5. Similar CPTs are determined for all the other nodes of the Bayesian Network, except the top level nodes.

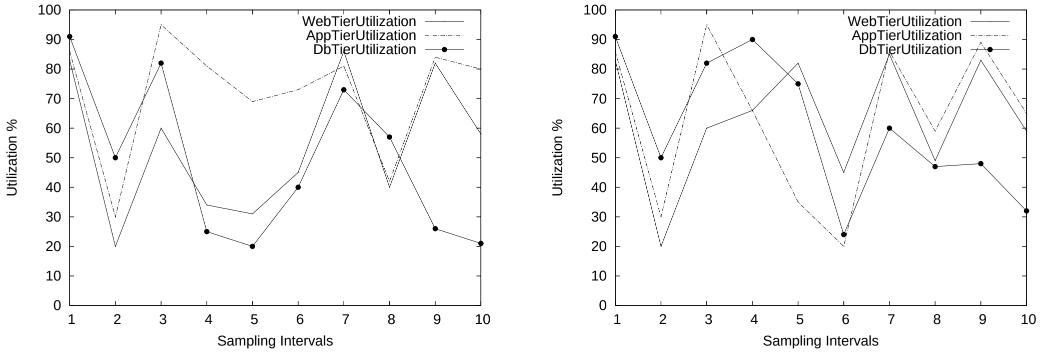
4 Performance Evaluation

4.1 Experimental Setup

To evaluate the session-based admission control strategies, we built a simulation model for a three-tier e-commerce site. Without being affected by the methods of implementations, the simulator can effectively evaluate the performance of the admission control strategies by itself. It consists of a customer generator, a session generator, a request generator, a web server, an application server,

Table 4: CPT for WebTierState Node

WTProcessingTime	WTUtilization	Normal	Underloaded	Overloaded
BT	BT	0	100	0
BT	NR	50	50	0
BT	AT	0	50	50
NR	BT	50	50	0
NR	NR	100	0	0
NR	AT	50	0	50
AT	BT	0	50	50
AT	NR	50	0	50
AT	AT	0	0	100



(a) Predicted utilizations with Blackbox. (b) Predicted utilizations with MBAC.

Figure 4: Predicted utilizations of a three-tier website.

and a database server.

As our previous work on session-based service differentiation on e-commerce servers [22], the session-based workload is generated following the guidelines provided by the TPC-W benchmark specification. A TPC-W workload session may belong to one of the three distinct traffic mixes, Browsing, Shopping and Ordering. Each of the workload mixes is characterized by different probability based navigational patterns. A session is created as a sequence of interactions for the same customer. For each session of a specific mix, the next interaction is determined by a state transition matrix that specifies the probability of moving from one interaction to another. The session time for the session and think time between the interactions are generated by an exponential distribution with a given mean [2]. The utilization and the processing time of each interaction is derived from the WIRT (Web Interaction Response Time) based on the observation that different mixes pose varying load on the tiers. As others in [12, 13], we assume that Browsing mix is database tier intensive, Shopping mix is application tier intensive and Ordering mix is web tier intensive. For example, for the requests of the Browsing Sessions, the database tier is the most intensive and the web tier is the

Table 5: Accepted sessions by Blackbox and MBAC.

Interval	By Blackbox	By MBAC
1	None	None
2	Any	Any
3	None	Ordering
4	None	Ordering
5	Any	Shopping
6	Any	Any
7	None	Browsing
8	Any	Any
9	None	Browsing
10	None	Any

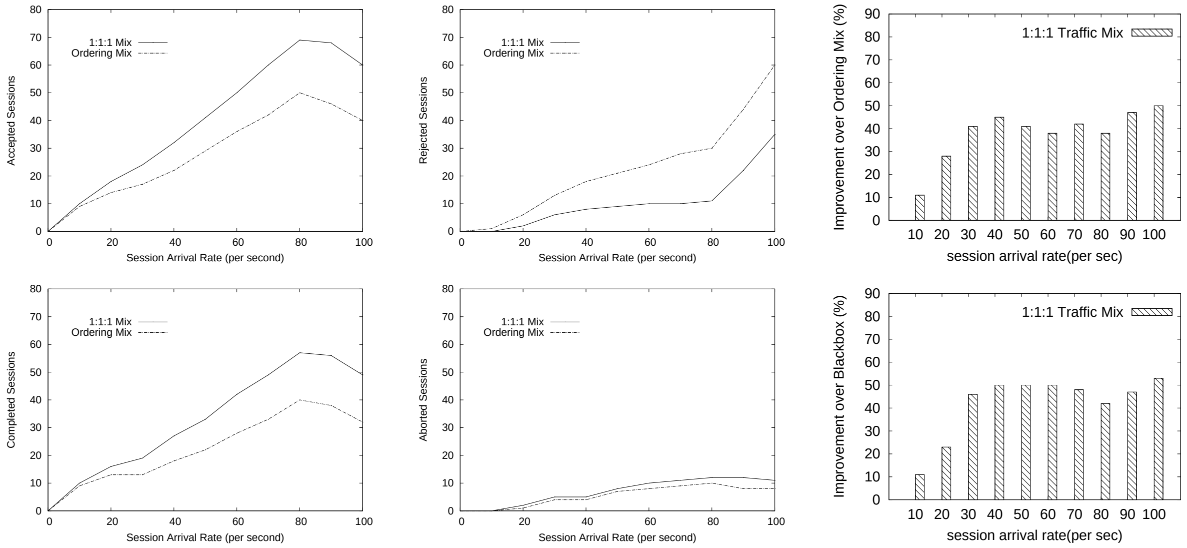
least intensive. The resource demand of a request on the web tier, application tier and database tier is set as 20%, 30% and 50% of the overall resource demand corresponding to its WIRT.

The Bayesian network used in the simulation was modeled with the Netica software [3]. 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. We next present the performance evaluation of admission control strategies.

4.2 Impact of Multi-tier Architecture on Session Admission Control

We first compare the Blackbox approach tailored from the SBAC approach [2] with the proposed MBAC (multi-tier measurement based admission control) approach. One main purpose is to demonstrate the impact of a multi-tier architecture on the effectiveness of session-based admission control. The workload in the experiment consists of an equal number of Browsing, Shopping and Ordering sessions. The experiment captured the predicted utilizations for the web, application and database tiers at the interval edges. Figure 4 shows the tier utilizations due to the Blackbox approach and the MBAC approach, captured in one period of 10 intervals.

At each interval, the admission control decision is made by comparing the predicted utilizations to a pre-configured threshold utilization values. For this experiment, the threshold utilization values for the web, application and database tiers is set at 80%. The admission control decision determines the type of the sessions that can be accepted for the next interval. For the Blackbox approach, if the predicted utilizations of *any* of the tiers exceeds 80%, no new sessions will be accepted in the next interval. For the MBAC approach, if the predicted utilizations of *all* of the tiers exceeds 80%, no new sessions will be accepted in the next interval. Otherwise, sessions of different traffic mixes



(a) Accepted sessions. (b) Rejected sessions. (c) Acceptance improvement.

(d) Completed sessions. (e) Aborted sessions. (f) Completion improvement.

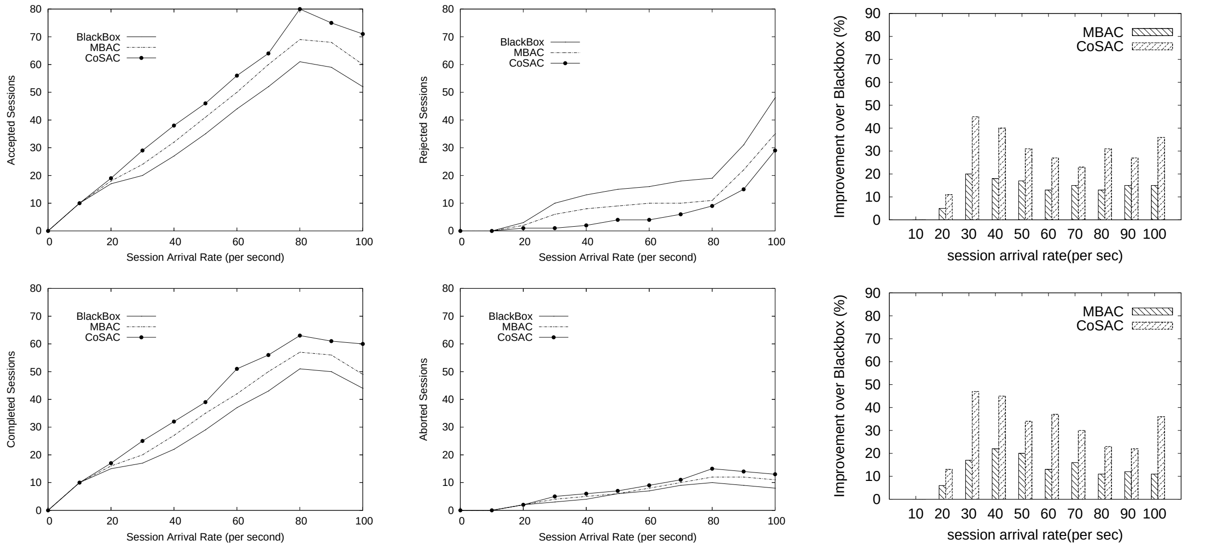
Figure 5: Throughput comparison with varying traffic mix.

will be accepted based on the utilization of the different tiers. Table 5 gives the session types that were accepted for the 10 intervals due to Blackbox and MBAC strategies.

Results clearly show that by the time the admission control decision is made at interval 1, the predicted utilizations of all the tiers exceed the threshold values. Thus, in both strategies, the admission control decision is to reject new sessions. At the edge of the interval 2, predicted utilizations of all the tiers fall below the threshold value, so new sessions of all traffic types will be accepted in the interval. So far the behaviors of Blackbox and MBAC strategies are similar. However, at the interval 3, only the web tier's predicted utilization is below the threshold. As the result, Blackbox approach rejects all the new sessions, while the MBAC approach accepts the Ordering sessions. No shopping and Browsing sessions will be accepted by MBAC approach as the utilization of the application tier and database tier are above the threshold. A similar observation can be found in intervals 4, 7 and 9. We observed that MBAC approach accepts sessions of different types while Blackbox approach rejects all new sessions in those intervals. Compared to the SBAC approach, the MBAC approach is able to accept more sessions, leading to more balanced utilization of the multiple tiers of the website.

4.3 Why Not Ordering Sessions First

One may argue that under heavy load conditions, intuitively only ordering sessions be accepted since they are more likely than other sessions to result in economic benefits for the e-commerce site. We executed experiments to support our counter-intuitive argument that accepting only ordering sessions under a heavy load condition is not the right way to utilize the resources in a multi-tier



(a)Accepted sessions. (b) Rejected sessions. (c) Acceptance improvement.

(d) Completed sessions. (e) Aborted sessions. (f) Completion improvement.

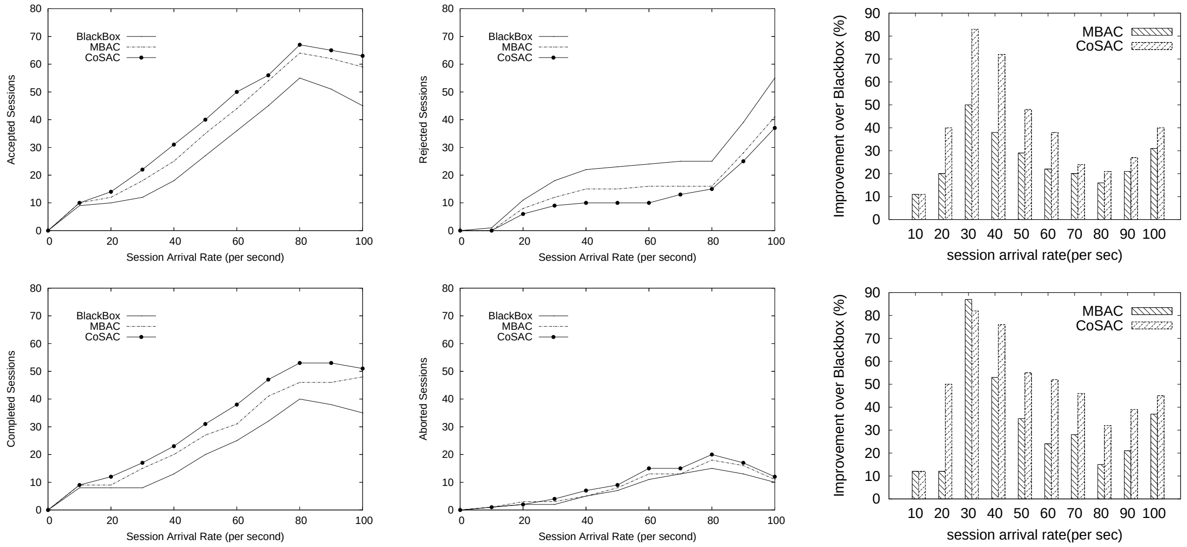
Figure 6: Impact of admission control strategies on session throughput (with 1:1:1 workload mix).

website. We applied a workload mix containing just Ordering sessions to the website with the MBAC admission control. We compare the throughput results with the experiment that applied a workload mix consisting of an equal number of sessions of all three mixes.

Figures 5(a)(b)(d)(e) show the experimental results of the number of accepted sessions, completed sessions, rejected sessions, and aborted sessions, respectively. Note that we represent the workload as the number of arrival sessions per second instead of a utilization percent. This is due the fact that there is no single utilization percent in a multi-tier architecture and the bottleneck tier shifts from time to time. The results show that accepting different traffic mixes can significantly increase the throughput, both as the accepted sessions and as the completed sessions. Figures 5(c)(f) depict the improvement in percentages. The rationale is that just accepting one type of sessions will overload one of the tiers sooner, leading to the system imbalance. On the other hand, accepting a mixture of different sessions will place more balanced utilizations across all three tiers. Thus the effective session throughput is increased significantly.

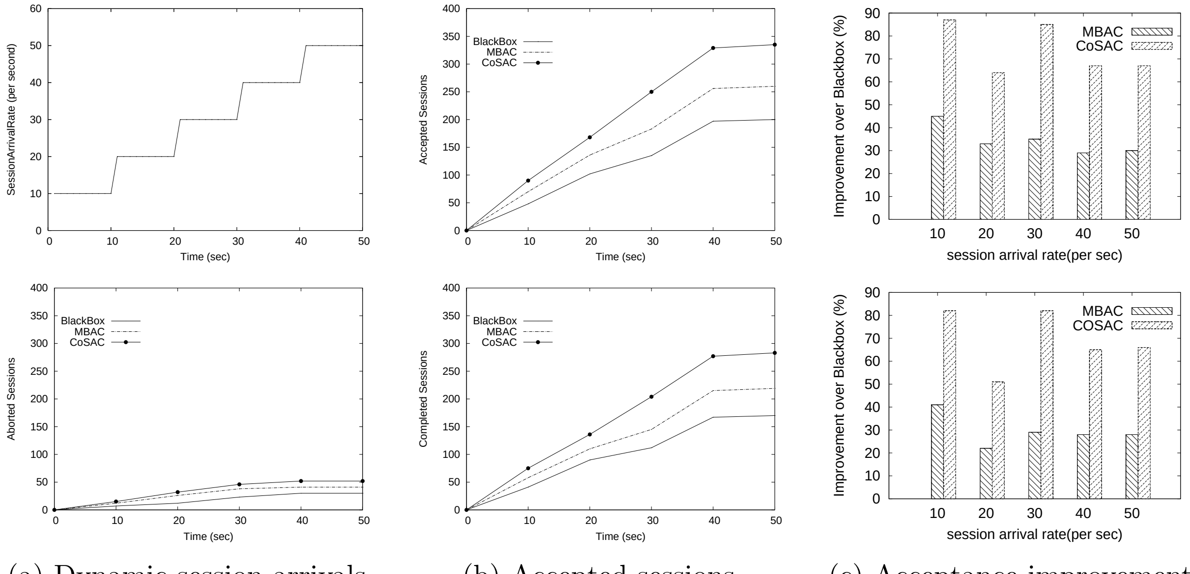
4.4 Impact of CoSAC on Throughput

We now evaluate the impact of CoSAC with a Bayesian network on the session throughput and compare it with the results due to the use of Blackbox and MBAC approaches. The workload consists of an equal number of Browsing, Shopping and Ordering sessions. We conducted experiments at different session arrival rates between 10 to 100 sessions/sec. Figures 6(a)(b)(d)(f) show the number of accepted, rejected, completed, and aborted sessions, respectively. Results demonstrate



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

Figure 7: Impact of admission control strategies on session throughput (with 3:2:1 workload mix).



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

Figure 8: Impact of admission control strategies on session throughput (with dynamic workload).

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 6(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 6(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.

Figures 7 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 6. 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 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 8(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 8(b)(d)(e) shows the number of accepted, aborted, completed sessions, and 8(c)(f) shows the improvements in percentage. We performed a wide range of sensitivity analysis. Due to the space limitation, we only presented those representative results. All results demonstrate the superior performance of CoSAC, which uses a Bayesian network to coordinate the session admission in a multiple tier website.

5 Conclusion

In this paper, we proposed two new session-based admission control approaches for multi-tier Internet applications. Both recognize the fact that the bottleneck in a multi-tier website shifts among tiers as client access patterns change dynamically. The existing session-based admission control approach (SBAC) based on the bottleneck tier is thus not effective as different client sessions impose different resource consumptions at the different tiers. The proposed MBAC approach is based on the dynamic measurement of utilizations at the different tiers and make admission decision pro-actively based on the session mix characteristics. The proposed CoSAC approach novelly uses a machine learning technique, a Bayesian network, to coordinate the admission decision based on the utilizations of the multiple tiers. Extensive simulation results based on TPC-W benchmark workload demonstrate the superior performance of the new approaches, particularly the CoSAC in terms of the effective session throughput.

Acknowledgement

This research was supported in part by U.S. National Science Foundation under Grant No. CNS-0720524. A preliminary version of the paper is to appear in the Proc. IEEE ICCCN'2009.

References

[1] J. Chen, G. Soundararajan, and C. Amza. Autonomic provisioning of backend databases in dynamic content Web servers. In *Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC)*, 2006.

[2] L. Cherkasova and P. Phaal. Session-based admission control: A mechanism for peak load management of commercial web sites. *IEEE Trans. on Computers*, 51(6):669–685, 2002.

[3] N. S. Corporation. Netica-j: Java netica api. http://www.norsys.com/netica-j.html.

[4] Y. Diao, J. L. Hellerstein, S. Parekh, H. Shaihk, and M. Surendra. Controlling quality of service in multi-tier Web applications. In *Proc. IEEE Int'l Conf. on Distribued Computing Systems*, 2006.

[5] Y. Diao, J. L. Hellerstein, S. Parekh, H. Shaihk, M. Surendra, and A. Tantawi. Modeling differentiated services of multi-tier web applications. In *Proc. IEEE MASCOTS*, 2006.

[6] S. Elnikety, E. Nahum, J. Tracey, and W. Zwaenepoel. A method for transparent admission control and request scheduling in e-commerce web sites. In *Proc. of the ACM WWW*, pages 276–286, 2004.

[7] J. Guitart, D. Carrera, V. Beltran, J. Torres, and E. Ayguade. Session-based adaptive overload control for secure dynamic web applications. In *Proc. IEEE Int'l Conf. on Parallel Processing*, 2005.

[8] T. Horvath, T. Abdelzaher, K. Skadron, and X. Liu. Dynamic voltage scaling in multitier web servers with end-to-end delay control. *IEEE Trans. on Computers*, 56(4):444–458, 2007.

[9] A. Kamra, V. Misra, and E. M. Nahum. Yaksha: a self-tuning controller for managing the performance of 3-tiered web sites. In *Proc. IEEE Int'l Workshop on Quality of Service (IWQoS)*, 2004.

[10] P. Lama and X. Zhou. Efficient server provisioning with delay guarantee on multi-tier clusters. In *Proc. IEEE Int'l Workshop on Quality of Service (IWQoS)*, 2009.

[11] S. Muppala and X. Zhou. CoSAC: coordinated session admission control for multi-tier Internet applications. In *Proc. IEEE Int'l Conf. on Computer Communications and Networks*, 2009.

[12] J. Rao and C. Xu. CoSL: a coordinated statistical learning approach to measuring the capacity of multi-tier Websites. In *Proc. IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS)*, 2008.

[13] J. Rao and C. Xu. Online measurement of the capacity of multi-tier websites using hardware performance counters. In *Proc. IEEE Int'l Conf. on Distributed Computing Systems*, 2008.

[14] Q. Tang, S.K.S. Gupta, and G. Varsamopoulos. Energy-efficient themal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. *IEEE/ACM Trans. on Parallel and Distributed Systems*, 19(11):1458–1472, 2008.

[15] B. Urgaonkar, G. Pacific, P. Shenoy, M. Spreitzer, and A. Tantawi. An analytical model for multi-tier Internet services and its applications. In *Proc. ACM SIGMETRICS*, 2005.

[16] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood. Agile dynamic provisioning of multi-tier Internet applications. *ACM Trans. on Autonomous and Adaptive Systems*, 3(1):1–39, 2008.

[17] D. Villela, P. Pradhan, and D. Rubenstein. Provisioning servers in the application tier for e-commerce systems. *ACM Trans. on Internet Technology*, 7(1):1–23, 2007.

[18] J. Wildstorm, P. Stone, E. Witchel, and M. Dahlin. Machine learning for on-line hardware reconfiguration. *International Joint Conferences on Artificial Intelligence*, 14(2):108–136, 2007.

[19] C.-Z. Zu. Scalable and Secure Internet Services and Architecture. *Chapman & Hall/CRC Press*, 2005.

[20] Q. Zhang, L. Cherkasova, and E. Smirni. A regression-based analytic model for dynamic resource provisioning of multi-tier Internet applications. In *Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC)*, 2007.

[21] J. Zhou and T. Yang. Selective early request termination for busy internet services. In *Proc. ACM WWW*, 2006.

[22] X. Zhou, J. Wei, and C.-Z. Xu. Resource allocation for session-based two-dimensional service differentiation on e-commerce servers. *IEEE Trans. on Parallel and Distributed Systems*, 17(8):838–850, 2006.