# Profit-aware Admission Control for Overload Protection in E-commerce Web Sites

Chuan Yue and Haining Wang
Department of Computer Science
The College of William and Mary
{cyue,hnw}@cs.wm.edu

***Abstract*—Overload protection is critical to E-commerce Web sites. This paper presents a profit-aware admission control mechanism for overload protection in E-commerce Web sites. Motivated by the observation [20] that once a client made an initial purchase, the buy-to-visit ratio of the client escalates from less than 1% to nearly 21%, the proposed mechanism keeps track of the purchase records of clients and utilizes them to make admission control decisions. We build two hash tables with full IP address and network ID prefix, which maintain the purchase records of clients in fine-grain and coarse-grain manners, respectively. We classify those clients who made purchases before as premium customers and those clients without prior purchase behavior as basic customers. Under overload conditions, our mechanism differentiates premium customers from basic customers based on the record hash tables, and admits premium customers with much higher probability than basic customers. In favor of premium customers, our mechanism maximizes the revenues of E-commerce Web sites. We evaluate the efficacy of the profit-aware mechanism using the industry-standard TCP-W workloads. Our experimental results demonstrate that under overload conditions, the profit-aware mechanism not only achieves higher throughput and lower response time, but also dramatically increases the revenue received by E-commerce Web sites.**

## I. Introduction

E-commerce has been rapidly growing in recent years. This rapid growth of E-commerce has imposed an ever-increasing workload on E-commerce Web sites, leading to a great demand for overload protection. An overloaded E-commerce Web site is swamped with numerous Web requests that are well beyond the system capacity. Without proper protection, system throughput drops quickly and the response time of those already-admitted requests increases dramatically to an unacceptable level. This in turn results in significant revenue loss to the overloaded E-commerce Web site. Latest studies have shown that 75% of visitors to a slow E-commerce site will never shop on that site again [17]. Being a temporary solution, simple over-provisioning mitigates the negative effect caused by overload but at very high cost. Moreover, simple over-provisioning cannot cope with flash crowds, the typical events that often overload Web sites [1], [10].

As effective approaches to overload protection in E-commerce Web sites, several admission control mechanisms have been proposed and developed [4], [5], [7]. The Web traffic of E-commerce is session-based, in which a session is defined as a sequence of temporally and logically related requests originated from the same client. Session integrity requires that once a request is admitted for processing, all the following requests within a session should be accepted. The importance of session integrity has been studied in [4], [5], and session-based admission control (SBAC) mechanisms has been proposed in [5]. More recently, Reward-Driven Request Prioritization (RDRP) mechanism [13] gives higher execution priority to the requests whose sessions are likely to bring more profit. However, these mechanisms focus on inter-request relationship within a session only, and none of them attempt to use the *inter-session* purchase record of a client for making admission decisions.

In this paper, we present a profit-aware admission control mechanism for overload protection in E-commerce Web sites. The key feature of our mechanism is to keep track of inter-session purchase records of clients and utilize them for admitting new sessions. The primary e-metric we use is the buy-to-visit (B2V) ratio, which captures the critical customer behavior of an E-commerce Web site [20]. The study in [20] shows that buyers without prior purchase record, i.e., who are making their first purchase, have a B2V ratio of less than 1%; however, once a client initiates a purchase, the B2V ratio of the client escalated to nearly 21%, approximating that one purchase for every five visits. This more than twenty times B2V ratio difference motivates us to design a profit-aware admission control mechanism, which differentiates premium customers who have made purchase(s) before from basic customers who have never made any purchase yet, and admits premium customers with much higher probability than basic customers under overload conditions. The major challenge to designing such a profit-aware admission control mechanism is how to classify customers in an efficient and reliable manner.

Utilizing IP address as a complementary but efficient customer classification method, we build two hash tables with full IP address and network ID prefix to maintain the purchase records of customers in fine-grain and coarse-grain ways, respectively. Under overload conditions, our profit-aware admission control mechanism gives much higher admission probabilities to premium customers than basic customers. In favor of premium customers, we can maximize the revenue of an overloaded E-commerce Web site. Based on the industry-standard TCP-W workloads, we evaluate the performance of the proposed profit-aware mechanism in our testbed. To highlight the importance of admission control and compare with current admission control mechanisms, we design and conduct four sets of experiments. The experimental results show that under overload conditions, the profit-aware mechanism not only achieves higher throughput and lower response time, but also dramatically increases the revenue received by E-commerce Web sites.

The remainder of this paper is structured as follows. Section II discusses the issues related to customers classification. Section III details the proposed profit-aware admission control mechanism. Section IV presents the experimental design and results based on the TCP-W workloads. Finally, Section V concludes the paper.

## II. Customer Classification

In general, E-commerce Web sites rely on login and password authentication as their primary method to explicitly identify a customer, and use cookies stored in a customer's Web browser as their secondary method to implicitly identify a customer. However, both methods have their own limitations. On the one hand, explicit customer authentication method may incur high shopping cart abandonment rate. One of the top E-commerce strategies of reducing shopping cart abandonment suggests to make registration and login optional [23]. This strategy has been widely adopted by most E-commerce Web sites and become the state-of-practice. On the other hand, although per-session cookies, which are stored in memory, have been widely used to identify the requests from the same client within a session, persistent cookies, which are used for inter-session and stored in disk, may be disabled or frequently cleared.

Using IP addresses to identify customers is straightforward and efficient, because an IP address comes with each Web request and no modification at client side is required. The major concern with using IP addresses for customer identification is the inaccuracy caused by dynamic addressing, NAT (Network Address Translation) boxes, and proxies. However, a recent study [3], which quantifies the effect of "edge opacity" on the IP-based customer identification, has shown us that most NATs are small serving fewer than 7 hosts and dynamic renumbering from DHCP (Dynamic Host Configuration Protocol) happens on the order of days.

Moreover, while DHCP may change the IP address of an end-host, it only changes the host ID but leaves the network ID intact. It is no surprise that network ID prefixes have been consistently stable [12]. Other recent measurements in DHCP [2], [16] have also shown that among hosts using DHCP, those frequently on-line hosts have the stable IP addresses. To this end, either long DHCP lease time is directly set, or the same IP address are continually re-assigned to the same host if a short DHCP lease time expires.

While proxies are much more likely to serve both larger and geographically diverse users, a set of techniques have been developed to detect such a proxy in real-time and enable the E-commerce Web site to make more informed decision on customer identification [3]. Furthermore, we can record such

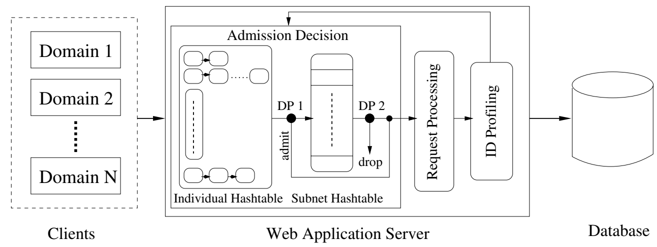



Fig. 1. Profit-aware admission control architecture.

an IP address of a proxy or a NAT box into the network ID prefix hash table as a special 32-bit long "network ID", since the corresponding purchase record also describes the aggregated customer behaviors behind the proxy or NAT box.

Although the number of IP addresses seen by an E-commerce Web site could be very large, our profit-aware admission control mechanism only needs to keep track of those IP addresses from which purchases have been made. Therefore, it is feasible for an E-commerce Web site to maintain the IP addresses of premium customers in a reasonable duration. Moreover, the network ID prefixes in the second hash table significantly condenses the IP address space into aggregated clusters. The memory efficiency of network ID prefix clustering has been shown in the study of hop-count filtering [9]. One noticeable example confirming the feasibility of our approach is that recently, Bank of America associates customers' on-line IDs with their frequently-used IP addresses to enhance on-line banking security [15].

## III. Profit-aware Admission Control

The proposed profit-aware admission control mechanism consists of two major modules: the ID profiling module and the admission decision module. The ID profiling module is always turned on to record customer purchase behaviors. When a payment is confirmed in the session, ID profiling module will record the client's IP address in the first hash table and update the corresponding entry in the second hash table. The admission decision module is in action only when the Web site is under an overload condition. The architecture of the profit-aware admission control mechanism is illustrated in Figure 1, in which a stream of session requests originating from multiple domains arrive at the E-commerce Web site.

### A. ID Profiling Module

The workload unit of E-commerce is a session. Each session consists of a sequence of various requests generated by a single customer during one visit to the Web site. The typical request types for an on-line shopper include browsing, searching, adding to the shopping cart, and making payment. The navigational pattern of a customer at the Web site can be described by Customer Behavior Model Graph (CBMG) [11]. With different transition probabilities, different CBMGs can be used to characterize different customer classes.

The internal structure of CBMGs could be very complicated and hard to precisely track for an E-commerce Web site. However, our ID profiling module does not need to track the internal navigational sequence of a customer session. This is the key difference between our mechanism and RDRP proposed in [13]. RDRP focuses on the inter-request structure within a session, thus relies on the accurate tracking of the internal structure of CBMGs. Our mechanism, however, keeps track of inter-session relationship for customer classification, and relieves the burden of tracking internal structure of CBMGs. Our ID profiling module records the IP address of a customer only when the customer confirms a payment. Because of this stateless property, the overhead induced by the ID profiling module is minor, although this module is always in active. The profiled ID information is maintained at two hash tables, which will be used for making admission decisions.

### *B. Individual Hashtable and Prefix Hashtable*

The key data structures of the profit-aware admission control mechanism are two hash tables. The first hash table, named the Individual Hashtable, keeps the full 32-bit individual IP addresses of premium customers. The second hash table, named the Prefix Hashtable, records the network ID prefixes of all customers. The Individual Hashtable records the most recent payment behaviors of individual IP addresses, providing fine-grain and short-term information for making admission decisions. The Prefix Hashtable captures the aggregated payment behaviors of network ID prefixes, providing coarse-grain and long-term information for making admission decisions. The IP addresses recorded by ID profiling module are maintained in these two hash tables and will be used by admission decision module.

Due to memory limitations, the Individual Hashtable cannot increase infinitely to accommodate every individual IP address. If the total number of elements in the Individual Hashtable reaches its capacity limit, some existing elements have to be evicted to save space for most recent customers. Therefore, the Individual Hashtable needs a replacement algorithm similar to those used in cache replacement. The basic requirements for our replacement algorithm include: those IP addresses from which purchases are received frequently or recently should not be replaced; and the replacement cost should be very low. To meet these requirements, we implement the clock replacement algorithm [6] in our Individual Hashtable.

In the Prefix Hashtable, a network ID prefix, instead of 32-bit full IP address, is maintained as the single entry of the table. For each network ID prefix that has been seen, the total numbers of committed purchases and completed sessions originated from it are recorded in the hash table. Therefore, we can compute the aggregated buy-to-visit (B2V) ratio of a network ID prefix based on the Prefix Hashtable. When a customer confirms a payment, its network ID prefix will be hashed into the Prefix Hashtable. If the entry for this network ID prefix is already there, its total number of committed purchases is increased by one. If the entry for this network ID prefix is not yet in the table, a new entry is created for the network ID prefix with its total number of committed purchases set to one. Once the network ID prefix is in the table, we will keep track of the total number of completed sessions for the network ID prefix as well.

```
Algorithm: Admission Decision
newIP = the IP address of the session
if ( newIP is in the Individual Hashtable )
    set its reference bit to 1
    admit this new session
else
    newNetID = the network ID prefix of the new session
    if ( newNetID is in the Prefix Hashtable )
        map its B2V ratio to a B2V ratio interval
        admitProb = the admission probability
            of the B2V ratio interval
    else
        admitProb = the minimum admission probability
            of all B2V ratio intervals
    endif
    if ( random() ≤ admitProb )
        admit this new session
    else
        reject this new session
    endif
endif
```

Fig. 2. The algorithm for new session admission decision.

### *C. Admission Decision Module*

The admission decision module is in action only when the Web site is overloaded. Admission decisions are made at two decision points DP1 and DP2, respectively, as shown in Figure 1. Under overload conditions, the IP address of a new session is first looked up in the Individual Hashtable. If the IP address is already in the table, the corresponding reference bit of the entry is set to one, and the new session is admitted into the system with probability one [1]. The above admission decision is made at DP1. However, if the IP address of the new session is not in the Individual Hashtable, the Prefix Hashtable will be looked up. If the network ID prefix of the IP address is in the Prefix Hashtable, we compute the admission probability of the new session based on the B2V ratio of the matched network ID prefix; otherwise, a minimum admission probability will be applied. For different network ID prefixes in the table, the larger the B2V values, the higher the admission probabilities. To facilitate admission control, the computation of admission probability is based on a certain number of B2V ratio intervals. For example, we can set $m$ (say 12) B2V ratio intervals, $I_1$, $I_2$, ..., $I_m$, with their corresponding B2V ratio ranges, $[0, \frac{1}{2^{m-1}}]$, $(\frac{1}{2^{m-1}}, \frac{1}{2^{m-2}}]$,..., $(\frac{1}{2},1]$. The admission probability of each B2V ratio interval is set proportional to its corresponding B2V ratio range upper bound. For any given network ID prefix, its B2V ratio is mapped into a B2V ratio interval, and then the admission probability of the network ID prefix is set to that of the B2V ratio interval. This admission decision happens at DP2. The complete admission decision algorithm is described in Figure 2.

[1] We assume that the capacity of an E-commerce Web site can support all its premium customers; otherwise, the Web site should increase its processing capacity or a probability less than one has to be applied.

## IV. Performance Evaluation

In this section, we first describe the experimental setup and the experimental design for evaluating the profit-aware admission control mechanism. Then, we present the experimental results based on the TPC-W workloads.

### A. Experimental Setup

The experimental setup includes three components: testbed, workload, and performance metrics, which are detailed in the following.

*1) Testbed Configuration:* Our testbed consists of three PCs connected by a NETGEAR 10/100Mbps fast Ethernet switch. The first PC is used as a database server, the second is used as a Web application server, and the third is used for client request generation. Each PC has a Pentium IV 2.8GHz CPU and 500MB memory, and runs SUSE Linux with kernel version 2.6.11. For the database server, we choose MySQL [22] because it is the most popular open source database and has been used by many popular E-commerce Web sites [21]. We run the current production release MySQL 5.0 as our database server. For the Web server, we choose Jetty [18], a fully-featured Web server for static and dynamic content. Jetty has also been widely used in commercial and open source products [19]. We use the current stable release Jetty-5.1 as our Web server.

*2) TPC-W Workload:* To fully evaluate the profit-aware admission control mechanism, the workload of our testbed follows the standard TPC-W [26] specification from the Transaction Processing Council (TPC). TPC-W exercises an online bookstore, which supports a full range of activities, such as multiple on-line sessions, dynamic page generation, and online transactions. In TPC-W, a Web interaction refers to a complete cycle of the communication between an Emulated Browser (EB) and the system under test. TPC-W simulates three different Web interaction mixes by varying the B2V ratio: browsing mix, shopping mix, and ordering mix. For these three mixes, the percentages of buy confirm Web interaction among all Web interactions are 0.69%, 1.20%, and 10.18%, respectively. We scale our TPC-W database to have 10,000 book items and 288,000 customers. We use the free available TPC-W implementation developed by PHARM team [25], which is written in Java and utilizes Java servlets to generate Web pages on the fly. The connection to the MySQL database is enabled by the MySQL JDBC connector. In our TPC-W testbed, the Jetty Web server uses a thread pool to process Web requests, and uses a JDBC connection pool to communicate with the database server.

*3) Performance Metrics:* Three performance metrics are used in this study: the number of Web interactions per minute, the number of payments per minute, and Web interaction response time. The number of Web interactions per minute quantifies the system throughput at the interaction level. The number of payments per minute is the index of the session integrity, which estimates the profit brought to an E-commerce Web site. For Web interaction response time, the TPC-W specification [26] has strict constraints for each type of Web interaction. It requires that during each measurement interval, at least 90% of Web interactions of each type should have Web interaction response time less than the specified constraint (in seconds). For example, a shopping cart interaction should have a Web interaction response time less than three seconds.

### B. Experimental Design

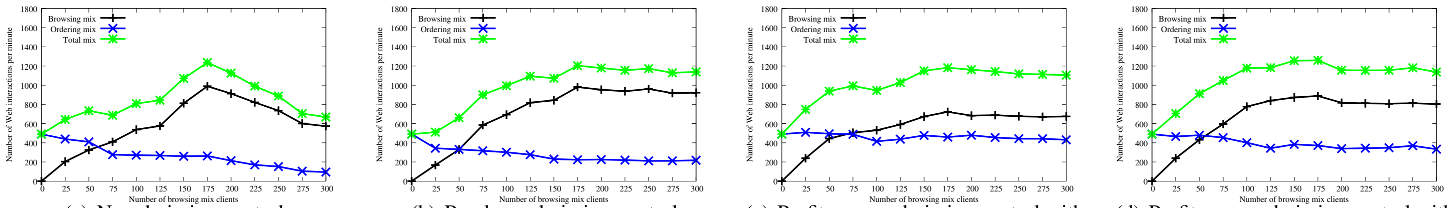
Similar to [7], in our experiments, system capacity is determined offline by using the incremental steps method [8]. The number of the emulated on-line browsers in each experiment approximates the load applied to the system. In general, there are two kinds of customers to an E-commerce Web site: basic customers who just browse and seldom order products, and premium customers who do order products in addition to browsing. The session scenario of a basic customer corresponds to the browsing mix of the TPC-W specification, while the session scenario of a premium customer corresponds to the ordering mix of the TPC-W specification.

To evaluate the effectiveness of the profit-aware admission control mechanism, we design four sets of experiments. Each set consists of 13 experiments with different load applied to the system. In the first set of experiments, there are no admission control or overload protection mechanisms. In the second set of experiments, we have a random admission control mechanism in the Web server. Here random means that the admission control mechanism is not aware of a customer's type, i.e., whether it is a basic customer or a premium customer, although the admission control system is session-aware. In the third set of experiments, we simulate the ideal case in which the profit-aware admission control mechanism has full knowledge of each customer's identity. In the fourth set of experiments, we simulate the case that the profit-aware admission control mechanism can only accurately distinguish 30% of the premium customers' identities. It means that the identities of premium customers captured in the Individual Hashtable are far from complete, and only those identified premium customers will be admitted with certainty. Our attempt is to show that the profit-aware admission control mechanism is still very effective even with the very limited (30% percentage) knowledge of premium customers. For each set of experiments, the number of on-line premium customers (ordering mix clients) is a constant 50, while the number of on-line basic customers (browsing mix clients) increases from 0 to 300, changing the system running status from normal to overload.

### C. Experimental Results

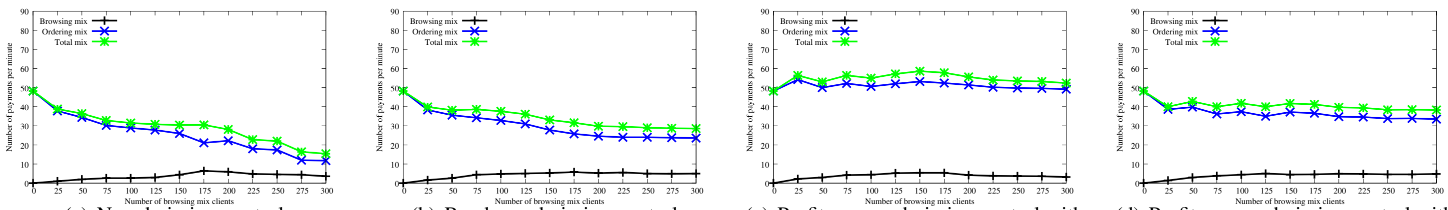
Normally, the bottleneck of an E-commerce Web site lies in its database server [7], [14]. This is also true in all our experiments, in which the resource bottleneck is either CPU or memory of the database machine. We monitor the system status of these machines through the Sysstat tool [24].

*1) Throughput in Web Interactions:* Figure 3 depicts the system throughput in terms of the number of Web interactions per minute as a function of the system load for the four sets (a, b, c, d) of experiments. The three curves in Figure 3 represent




(a) No admission control
(b) Random admission control
(c) Profit-aware admission control with 100% knowledge
(d) Profit-aware admission control with 30% knowledge

Fig. 3. Throughput in terms of number of Web interactions per minute as a function of the system load for the four sets of experiments. In each sub-figure, the Browsing mix curve stands for the throughput for basic customers, the Ordering mix stands for the throughput for premium customers, and the Total mix curve is the overall throughput for the two kinds of customers.




(a) No admission control
(b) Random admission control
(c) Profit-aware admission control with 100% knowledge
(d) Profit-aware admission control with 30% knowledge

Fig. 4. Throughput in terms of number of payments per minute as a function of the system load for the four sets of experiments. In each sub-figure, the Browsing mix curve stands for the throughput for basic customers, the Ordering mix curve stands for the throughput for premium customers, and the Total mix curve is the overall throughput for the two kinds of customers.

the throughputs for basic customers, premium customers, and all customers, respectively.

As shown in Figure 3(a), if there is no admission control, the database machine is busily swapping pages, and the responses sent back to the front Web server are dramatically reduced in the highly overloaded cases. In the case for random admission control, which is shown in Figure 3(b), with the increasing of the number of on-line browsing mix clients, the throughput for the ordering mix will again decrease due to the resource competition from the browsing mix. However, the throughput degradation is slightly slower than before.

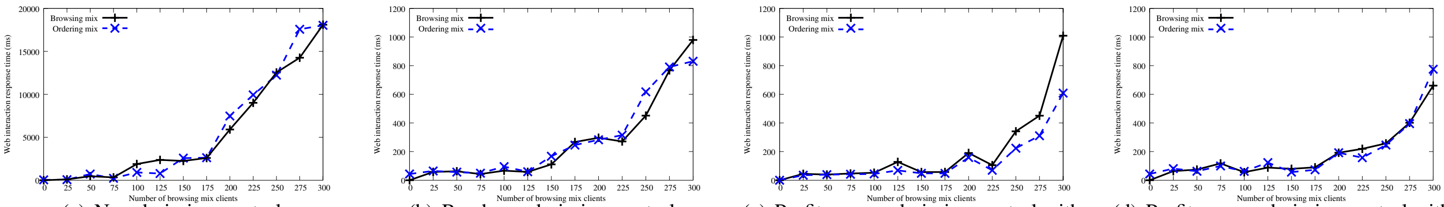
Figure 3(c) illustrates the experimental results for profit-aware admission control with 100% knowledge. It clearly demonstrates that the profit-aware admission control mechanism accepts all the premium customers' session requests, and only admitting the basic customers' session requests based on the remaining of system capability. In the fourth set of experiments, as shown in Figure 3(d), the total mix throughput is similar to those of random admission control and profit-aware admission control with 100% knowledge. The throughput for the ordering mix remains at about 350 Web interactions per minute after the number of browsing mix clients is 175. Although this is lower than the 450 Web interactions per minute of the ideal case in Figure 3(c), it is still much better than the 220 Web interactions per minute of the random admission control in Figure 3(b).

*2) Throughput in Payments:* We use the throughput in terms of number of payments per minute to clearly estimate the financial benefits brought by different admission control mechanisms. Figures 4 (a), (b), (c) and (d) illustrate the corresponding results with respect to the four set of experiments, respectively. It is evident that for all the four sets of experiments, the overall (total mix) throughput in terms of number of payments per minute is mainly attributed to the ordering mix.

With the increase of browsing mix clients, for both no admission control shown in Figure 4(a) and random admission control shown in Figure 4(b), their overall payment throughputs consistently degrade, while the latter drops more slowly than the former. Figure 4(b) shows that the current session-based admission control mechanism falls short in its capability of protecting premium customers and maximizing the profits of E-commerce Web sites.

For profit-aware admission control with 100% knowledge shown in Figure 4(c) and profit-aware admission control with 30% knowledge shown in Figure 4(d), both can achieve a relatively stable and much higher throughput in payments. No matter how heavy the system load is, profit-aware admission control can always give high priority to the session requests from premium customers, providing the maximal profit for an E-commerce Web site.

*3) Web Interaction Response Time:* We present the Web interaction response times under different admission control mechanisms in Figure 5. As shown in Figure 5(a), without admission control, the Web interaction response time exceeds three seconds when the number of browsing mix clients reaches 175, and then increases dramatically up to about 18 seconds. In contrast, for the cases with admission control mechanisms, which are shown in Figures 5(b), 5(c), and 5(d), the Web interaction response time grows with the increase of the system load but is always less than one second. We can also see that the Web interaction response time for ac-




(a) No admission control (b) Random admission control (c) Profit-aware admission control with 100% knowledge (d) Profit-aware admission control with 30% knowledge

Fig. 5. Web interaction response time (ms) as a function of the system load for the four sets of experiments. In each sub-figure, the Browsing mix curve stands for the response time for basic customers, the Ordering mix curve stands for the response time for premium customers.

cepted requests under different admission control mechanisms are similar. This is because once admission control prevents the database machine from thrashing, Web requests can be processed in an effective manner. In addition, there is no noticeable difference on the Web interaction response time between the two types of customers. This can be explained by the fact that although browsing mix clients seldom order products, their Web interactions involve many database transactions such as searching new products or viewing product details.

## V. Conclusions

In this paper, we propose a profit-aware admission control mechanism for overload protection in E-commerce Web sites. It is motivated by the facts that once a client made an initial purchase, the buy-to-visit ratio of the client escalates from less than 1% to nearly 21%. Our mechanism uses IP addresses to identify customers, and maintains their purchase records to make admission control decisions. Our mechanism builds one individual IP hash table and one network ID prefix hash table to keep track of customer purchase records in fine-grain and coarse-grain manners, respectively. Under overload, our mechanism favors premium customers over basic customers so that premium customers can be protected, which maximizes the E-commerce Web site's revenue.

Through extensive simulation experiments on the TPC-W based testbed, we demonstrate that during an overload period, the profit-aware admission control mechanism can not only achieve higher throughput and lower response time, but also significantly increase the revenue of an E-commerce Web site. Moreover, the proposed mechanism, which is also session-aware, is complementary to the existing session-based admission control mechanisms and can be easily deployed at an E-commerce Web site.

## References

[1] M. Arlitt and T. Jin, "Workload characterization of the 1998 world cup web site," *Network, IEEE*, vol. 14, no. 3, pp. 30–37, 2000.

[2] V. Brik, J. Stroik, and S. Banerjee, "Debugging dhcp performance," in *Proc. the 4th ACM SIGCOMM IMC*, October 2004.

[3] M. Casado and M. J. Freedman, "Peering through the shroud: The effect of edge opacity on IP-based client identification," in *Proc. USENIX/ACM NSDI'2007*, April 2007.

[4] H. Chen and P. Mohapatra, "Session-based overload control in qos-aware web servers," in *Proc. IEEE INFOCOM'2002*, June 2002.

[5] L. Cherkasova and P. Phaal, "Session-based admission control: A mechanism for peak load management of commercial web sites," *IEEE Trans. Comput.*, vol. 51, no. 6, pp. 669–685, 2002.

[6] F. J. Corbato, "A paging experiment with the multics system," *MIT Project MAC Report MAC-M-384*, 1968.

[7] 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. the 13th international conference on World Wide Web*, May 2004.

[8] H.-U. Heiss and R. Wagner, "Adaptive load control in transaction processing systems," in *Proc. the 17th International Conference on Very Large Data Bases*, September 1991.

[9] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: An effective defense against spoofed DDoS traffic," in *Proc. ACM CCS'2003*, October 2003.

[10] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: characterization and implications for cdns and web sites," in *Proc. the 11th international conference on World Wide Web*, May 2002.

[11] D. Ménasce, V. Almeida, R. Fonseca, and M. Mendes, "A methodology for workload characterization of e-commerce sites," in *Proc. ACM Conference on Electronic Commerce*, November 1999.

[12] S. Y. Qiu, P. D. McDaniel, F. Monrose, and A. D. Rubin, "Characterizing address use structure and stabillity of origin advertizement in interdomain routing," in *Proc. the 11th IEEE Symposium on Computers and Communications*, June 2006.

[13] A. Totok and V. Karamcheti, "Improving performance of internet services through reward-driven request prioritization," in *Proc. the 14th IEEE International Workshop on Quality of Service (IWQoS)*, June 2006.

[14] Q. Zhang, A. Riska, and E. Riedel, "Workload propagation – overload in bursty servers," in *Proc. 2nd International Conference on Quantitative Evaluation of Systems (QEST)*, September 2005.

[15] "Bank of America," http://www.bankofamerica.com, 2006.

[16] "DHCP Leases, Lease Length Policies and Management," http://www.tcpipguide.com/, 2006.

[17] "E-Commerce facts posted by ZDNet Research," http://blogs.zdnet.com/ITFacts/index.php?cat=16&paged=2, 2007.

[18] "Jetty Java HTTP Servlet Server," http://jetty.mortbay.org/jetty/index.html, 2006.

[19] "Jetty Powered Commercial and Open Source Products," http://www.mortbay.com/mortbay/powered.html, 2006.

[20] "Measure Twice, Cut Once - Metrics For Online Retailers," http://www.techexchange.com/thelibrary/online_retail_metrics.html, 2006.

[21] "MySQL Customers by Industry," http://www.mysql.com/customers/, 2006.

[22] "MySQL: The World's Most Popular Open Source Database," http://dev.mysql.com/, 2006.

[23] "Reducing Lost Shopping Carts," http://beyondthebasket.org/blogs, 2006.

[24] "Sysstat Project," http://freshmeat.net/projects/sysstat/, 2006.

[25] "TPC-W Java Implementation by PHARM team," http://tpcw.deadpixel.de/, 2006.

[26] "Transaction Processing Performance Council Benchmark W (TPC-W)," http://www.tpc.org/tpcw/default.asp, 2006.