Admission Control and Request Scheduling for Secured-Concurrent-Available Architecture

N. Harini
Department of Computer Science and Engineering,
Amrita University,
Coimbatore, India.

T.R Padmanabhan, PhD.
Department of Information Technology,
Amrita University,
Coimbatore, India.

ABSTRACT

The Internet society is continuously growing and the need of performance studies in this field is essential in order to obtain better throughput for the Web users. Moreover, Internet applications and clients have varied service expectations and demand provisioning of different levels of QoS to multiple traffic classes on the Internet. Meeting client QoS expectations prove to be a difficult task for E-Commerce service providers, especially when web servers experience overload conditions that cause increased response time, request rejections, user frustration, lowered usage of the service and reduced revenues. A recently proposed MLF (Multi Layered Filtering) framework manages potential workload; it also guaranties data freshness for all transactions that complete within their deadlines without service differentiation using request filtering and admission control schemes. In this paper, an improved request classifier and class-based admission control scheme called **Multi-phase Admission Control (MPAC)** that can be integrated with the MLF framework to prioritize user requests into several service classes according to their importance, and offer differentiated quality of service in terms of minimizing the frustration of premium users is presented and evaluated. The performance evaluations carried out confirm that the scheme can significantly boost the reward for having serviced a request while providing better QoS to clients.

General Terms

Admission control, Differentiated service, QoS

Keywords

Multi layered filtering, Multi phase admission control, Differentiated Service, QoS, Request Classification.

1. INTRODUCTION

The extraordinary growth of the World Wide Web, the rise of modern Internet services and the characteristics of these services have given rise to a number of new design challenges. Many of the Internet services like E-Commerce, e-mail, on-line news, social networks, etc. have become a fundamental resource and are considered vital by many people. At the same time, these services are subject to massive variations in workload, which happen over a variety of time scales. That is, the client population for these websites (load on major newspaper/TV site when big news breaks, Share market's black Tuesday on 27th the Feb 2007, etc.) is characterized by large peaks in access rates during these events. The critical nature of these on-line services and the ever-increasing interest of server administrators to maximize their client satisfaction while efficiently using existing resources, increase the complexity of performance requirements. A recent work reported MLF[1] a practical (secured-concurrent-available) end-to-end framework based on admission control policies achieves robust performance on a wide range of Internet services subject to huge variation in load. In this paper a differentiated service based admission control strategy with an enhanced request classification scheme called Multi Phase Admission Control (MPAC) that can be integrated with MLF to bring more profits to the service that particularly benefits E-Commerce is proposed. The proposed scheme minimizes request processing time and maximizes the reward attained by processing requests that are likely to bring more profits to the service that particularly benefits E-Commerce websites. Simulations and experimentation show that differentiation in service results in a considerable improvement of performance in terms of reward earned for servicing the request and the same is detailed in section IV. The rest of the paper is organized as follows. Section II discusses the literature relevant for the motivation of the work. Sections III and IV analyze the performance of MPAC approach which focuses mainly on E-Commerce. Final conclusions are presented in section V.

2. BACKGROUND STUDY

As Internet services become more popular and pervasive, a serious problem that arises is managing the performance of services under intense overload. Internet services experience huge variations in service load, with bursts coinciding with the times that the service has the most value. Unfortunately, in traditional operating system designs, common models of concurrency and replication protocols are not integrated to provide graceful management of such peak loads. A recent work reported, MLF scheme addresses the issues of overload and staleness effectively and ensures that more number of legitimate requests is answered with complete data through control mechanisms that offer adaptive load shedding, improved availability, and secure transactions using a multi layered filtering. However, Increased competition, heightened customer expectations and the growth witnessed by electronic commerce demand for methods that provide differentiated classes of service for Internet traffic, to support various types of applications, and specific business requirements. For example, scenarios like online shopping where the QoS is evaluated in terms of the requests that bring profits to the service provider, need differentiated access to web servers through enhanced mechanisms that perform request classification and exercise admission control so that a certain class of requests (e.g. buy requests) gets precedence over other class of requests (e.g. browse requests).

2.1 Overview – MLF Scheme

MLF is a practical end-to-end framework based on admission control policies to handle massive concurrency in web servers. The hallmarks of the MLF scheme include providing security, concurrency and availability. The scheme offers resistance to peak loads by filtering malicious requests at an initial stage thus avoiding bottlenecks. Filtering is done using CAPTCHA test and then by a detection module that detects the DDoS attacks far ahead of servers. The MLF model categorizes resources into two types namely, resources that may not be replicated due to security concerns (e.g. building structure reports, intelligence, etc) and those that may be replicated (E-Commerce, News articles, Game scores etc). This categorization is essential as the availability is closely affected by the number of replications. The architecture does not require over-provisioning of resources to provide better QoS. It brings more reward for having serviced in terms of minimized user frustration and request rejection rates at times of emergency and its integration with replication protocol ensures completeness of the data made available. For a detailed understanding [1] may be referred.

2.2 Peak Load Management

The fast-growing number of users and the increasing pace of information exchange, bring many big challenges to the Internet server designs. One of the biggest concerns for Internet service providers is the service performance of server systems when enormous unexpected concurrent client requests are added onto the systems during big events. A recent worldwide event happened on 27 Feb 2007. In this share market's black Tuesday, the global stock market steeply plunged around 5%; many electronic stock trading sites worldwide clashed for hours because of unexpected volume surges and the heavy sell-off, which sent a big warning of how vulnerable the market structure is to system glitches and data backlogs[6]. A heavy workload may induce server thrashing and service unavailability.

In E-Commerce applications, such server behavior could translate to sizable revenue losses. For instance, [5] estimates that between 10 and 25% of E-Commerce transactions are aborted because of slow response times, which translates to about 1.9 billion dollars in lost revenue. For managing the multi fold increase in number of clients the available solutions are over-provisioning of resources or use of an appropriate concurrency model with a suitable admission control strategy to support more connections. Several different server architectures have been proposed for managing high levels of server concurrency. However architecture like MLF which provides high concurrency without sacrificing availability is wanting for E-Commerce as there is a clear need for data consistency and high availability.

2.3 Distributed Denial of Service (DDoS) mitigation

All of the benefits that the Internet offers including support for the most basic and essential services, are subject to disruption by Internet-based cyber-attacks. According to [2], a mere 171 vulnerabilities were reported in 1995, which boomed to 7236 in 2007. Distributed denial-of-service (DDoS) attacks create a serious threat to network security. The increase in Internet-based transactions offers new opportunity for hackers to disrupt business operations with DDoS attacks. Organizations not adequately protected risk losing customers, revenue, and their good reputations. A series of DDoS attacks occurred in February 2000 to considerable media attention, resulting in higher packet loss rates in the Internet for several hours [3]. The vulnerability of the Internet to DDoS attacks has driven a large number of research efforts on DDoS mitigation. Although a lot of methodologies and tools are devised to detect DDoS attacks and reduce the damage they cause, most of the methods fail to simultaneously achieve efficient detection with a small number of false alarms and real-time transfer of packets. There are many well-known methods for classification like SVM, NN, fuzzy logic, and rough set. Hoai-Vu Nguyen and Yongsun Choi in their study [4] have identified an approach that detects DDoS attacks using k-NN classifier at a very early stage within a short time.

2.4 Admission Control Mechanisms

The admission control of MLF scheme is based on the policies defined by the administrator on input parameters (Max clients, Queue length, Arrival rate etc set by him initially). The architecture can be generalized to suit E-Commerce applications by integrating the admission control policies with a reward function that provides metric based differentiation and maximizes the profit earned for having serviced a certain class of requests(for an online function the number of items sold determines the profit earned). Research in this area has identified some key approaches to face with overload, such as admission control (per request, per session), request scheduling, service differentiation, service degradation or resource management. Session based admission control strategy which is widely adopted by researchers is chosen for implementation and the same was found to offer improved performance.

2.5 Summary of Findings

System overload is a common situation, and its commonness is growing along with the popularity of Internet services that increasingly demand more resources. Internet based E-Commerce services grow at a rapid pace. This imposes an ever-increasing workload on E-Commerce Web sites, leading to a great demand for overload protection. Improper defense leads to drop of system throughput and increased response time of those already-admitted requests. This results in significant revenue loss to the overloaded E-Commerce Web site. Studies show that about 75% of visitors to a slow E-Commerce site will never shop on that site again [8]. Resource over-provisioning mitigates the negative effect caused by overload but at very high cost. Moreover, simple over-provisioning cannot handle typical events like flash crowds that often overload Web sites [7]. A number of approaches for overload protection in E-Commerce Web sites have been proposed and developed by researchers. Many of them require extensive changes to the server or operating system [9] and lack integration of concurrency and replication models in the architecture. With increase in commercialization of Internet and E-Commerce applications becoming sophisticated in their data needs, there is a clear need for architecture with refined request admission control policies that provide differentiated QoS in terms of servicing a request with fresh data, in spite of the variability and diverseness among incoming requests.

2.6 Problem statement

The primary goal is to enhance the performance of MLF scheme and enable real world applications to take advantage of this functionality. A scheme that performs admission

control with enhanced request classification called Multi Phase Admission Control (MPAC) that can be integrated with MLF scheme to maximize the reward earned for having serviced a particular class of requests is proposed in this paper and the same evaluated using a simple analytical model. The proposed strategy operates in two phases. In the first phase it reduces the probability of successful attacks by using a filter relatively more intensive than one in MLF. In the second phase it schedules the requests that are likely to bring more service profits while others are taken up with less priority as the case may be.

3 Proposed Scheme – Multi Phase Admission Control

With the innovations made in modern electronic commerce, the number of users exploiting these services has grown exponentially As the demand for the E-Commerce service increases, it becomes crucial to have sufficient infrastructure for accommodating a large number of simultaneous users and support various services for a sustained period of time due to the unpredictability of the number of users utilizing this system. Realistically, as the resources of the service provider are less than that required to meet the peak requirement, it becomes necessary to employ appropriate admission control measures to accept a premium subset of incoming requests and service them within their QoS bound and reject the remaining requests.

3.1 Model and assumptions

The model assumes an input set of n requests, where each request is represented in terms of basic attributes as R {arrival Time (a_i), service Time (s_i), responseTime Bound (tb_i), capacity(C(overall),c_i (per request)), service Class(sc_i), State(S_i) }. The overall objective is to service maximum number of reward fetching requests (x_i).

Objective function: Max $\sum i = 1\ to\ n(R\sum t = 1\ to\ tmax(x_{it}))$
Subject to constraints $\sum i = 1\ to\ n\ (\boldsymbol{ci}R(x_{it})) \leq C$

Where xit =	1	if request i is scheduled at time t and completed within time bound i.e. t+si-ai) <= tbi
	0	otherwise

Requests are scheduled by quantifying the service to a customer such that the request processing time is minimized and the overall revenues are maximized while preserving ease of implementation. Reward is characterized by attributes (s_i, sc_i, S_i) and is earned if $x_{it} = 1$ and no reward is earned if $x_{it} = 0$.

Service Level Agreements (SLAs)

These mechanisms are needed to aid in management and diagnosis of end-to-end services. Service providers use SLAs as a means of specifying service level attributes that are offered to their customers. This implies that it is necessary for a provider to meet their service level obligations, and to enable providers to manage their infrastructure conforming to those agreements. It allows a service provider to offer verifiable and meaningful service behavior to their customers. Providers can offer customers the capability to automatically verify the current service behavior against the guarantees, by exposing the values of service parameters as agreed upon in the contract. The template framework for an SLA which forms the basis for our implementation is given here. The SLA framework can be seen to spell out the scope of service providers' allotment to the E-Commerce in terms of resource capacity and time commitment.

Contract: E-Commerce System

[

{

Service : Classification of Customers

Customer Class = {Premium, Ordinary, New}

Inter-session States = {Home, Browse, Item, Addcart,

BuyReq, BuyConfirm}

}

{

Service : Processing Requests (Peakload)

{Availability > $minAvailability;

Time-Bound < $maxDelay;

Throughput> $minThroughput;

Utilization < $maxUtilization;

Weight_Adjustment(Forward) = $weight (Positive)

Weight_Adjsutment(Backward) = $weight

(zero,negative)

}

]

3.2 Phase I

The MLF scheme offers resistance to peak loads by filtering malicious requests at an initial stage using two filters to classify incoming requests as legitimate and malicious. Although the FCM algorithm used in [1] does efficient classification of requests for the purpose of DDoS defense it is relatively slow in examining real time network traffic. Applications like E-Commerce require clustering be applied to the data stream. Amongst algorithms available for clustering [4] identifies k-NN as a better classifier for easier implementation, short time computation and high accuracy reasons. The detection module at the public server was replaced with the k-nearest neighbor (k-NN) method. In line with the paper the following parameters were used for classification: entropy of source IP address, entropy of port number, entropy of destination IP address, entropy of destination port number, entropy of packet type, occurrence rate of packet type and the number of packets. The use of k-NN method helped in early detection of malicious packets within short duration of time and achieve dimensionality reduction in terms of the number of parameters (from 16[1] to 7).

3.3 Phase II

Legitimate requests are forwarded to access points by the public server that provide differentiated access to web servers through enhanced mechanisms that provide revenue-aware admission control scheme for overload protection. The major challenge to designing such a revenue-aware admission control mechanism is classifying customers in an efficient and

reliable manner. Such a classification scheme has been evolved here. The key feature of the mechanism is to assign an initial weight based on the past data and update the same by keeping track of inter-session purchase records of clients and utilize them for admitting new sessions. Request profiling module performs real-time monitoring of client requests to extract parameters of service usage and to maintain the histories of session requests; it also assigns request priorities that in turn influence queuing behavior for various application server resources. This is illustrated in figure 1

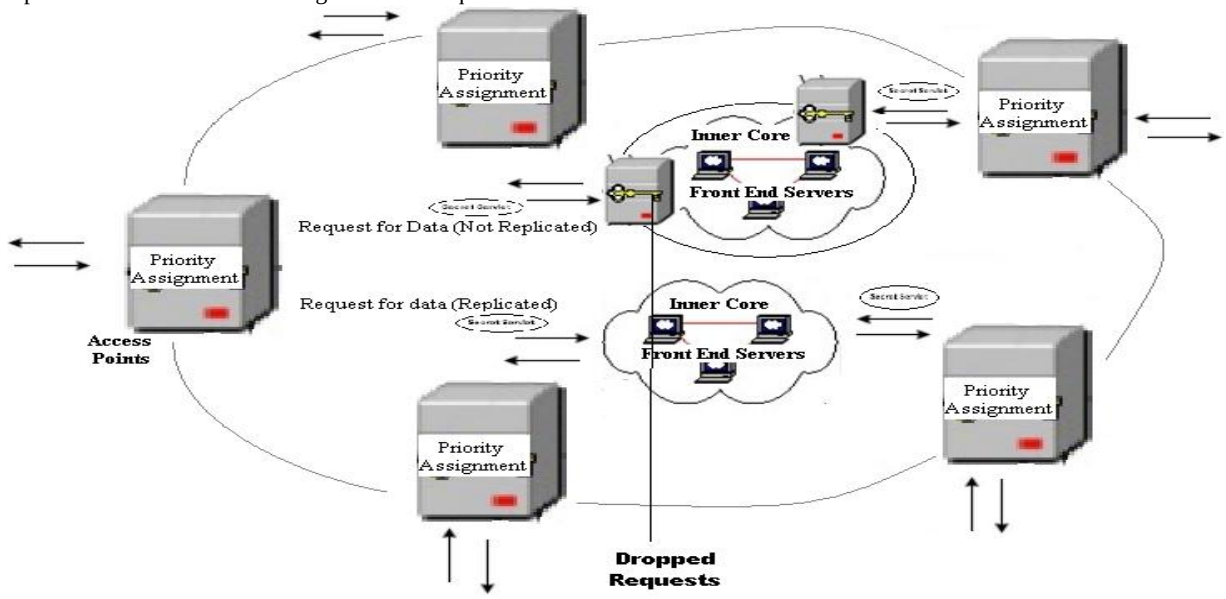

Fig 1: Request Processing in Stage II of MLF

3.3.1 Overall Strategy

A table of individual IP addresses of all customers who have completed at least one purchase is maintained. When a customer confirms payment the table is scanned for the corresponding entry and if exists the total number of purchases is increased by one else a new entry is created with its total number of purchases set to 1.

For each class assign an initial weight

/*(e.g. Premium = 3, Ordinary = 2 and New = 1)*/

For each state assign an initial weight such that the

weight increases as one makes a forward transition

/*e.g. home = 0.1, browse = 0.2 etc */

For each state initialize the expected reward and Cost

incurred as {Reward attained for processing a request in

that state and

Cost incurred for processing a request in that state}

Weight = Ini_weight;

Curr_sess_value = Weight of the first state in the

sequence;

/* Prev_sess_value = Weight of the first state in the

sequence;/*

tlimit is the time limit for which the behavior of the

customer will be observed

For time = 1 to tlimit

{

For each state change

{

Observe the state transitions and the timings involved

and update the weight

/*(e.g.)

if Prev_sess_Value > Curr_sess_value

then

weight = weight – (Prev_sess_value – Curr_sess_value)

elseif

Prev_sess_value < Curr_sess_value

then

weight = weight + (Prev_sess_value – Curr_sess_value)

elseif

timeout ((T + Si(Tq+Ts) – Ai) > Tb_i)

then

drop the request;

where T is the time spent in stage 1 of MLF Si is the service time availed so far by request i (it includes the time spent in the queue and the actual service time), Ai is the arrival time of the request and Tb_i is the time bound*/

}

Compute Expected Reward and Priority as per the reward function specified by the site owner

Queue in the request in priority queue

Invoke the scheduler

}

3.3.2 Initial assignment of weight

Each incoming request is assigned an initial weight computed based on the number of purchases and number of visits already made (obtained from the history of records).

3.2.3 Computation of Expected reward and priority

Find the membership value of the current session with sessions in the previous history. Select those records where the membership values are above a threshold value [9].

{e.g. let the number of such selected records = I}

Predict the session benefit

/*Exp_Reward = Exp_rew{Curr_sess}i X

$$\frac{\sum_{k=1}^{I} rew_{att(Curr\_sess)k} * mem_{val}\{Curr\_sess\}i \text{ to } k}{I}$$

$$\text{Cost\_Inc} = \text{Exp\_cost}\{\text{Curr\_sess}\}_i \text{ X } \frac{\sum_{k=1}^{I} Cost\_inc_{(Curr\_sess)k} * mem_{val}\{Curr\_sess\}i \text{ to } k}{I} */$$

Compute priority as

/*Priority = Initial Priority + (Exp_reward/Exp_cost)*/

3.3.4 Pseudo code for the scheduler

Let the Arrival Rate at public server = Ar(Pub) and

Service Rate = Sr(Pub).

If Ar(pub) <=Sr(pub) then

system is stable

Else

unstable(overloaded).

The Utilization of a server is computed as

U = Ar(pub) * Ts(pub).

Consider a set of n requests arriving at the public server

at time T{1 to Tmax}.

Let each request Xi be defined with parameters Arrival time of a request = Ai, Service time = Si,Time bound Tbi(pub)

/* time within which the servicing should be completed

*/.

The utilization factor(Ar *Sr) for all the access nodes is computed and recorded by the monitor (Spl Server – A dedicated public server)

If (T + Si(Tq+Ts) – Ai) <= Tbi

Then

Schedule the request

otherwise

drop the request

/*where Tq is the time delay in queues (ensures that servicing the request is profitable) */

Forward the scheduled request to the server being utilized less (obtain utilization factor from the monitor). This improves average server utilization compared to other schemes like Round-robin.

3.2.5 Computing Reward Earned

It is assumed that the E-Commerce website has capacity to support all its premium customers. It is the service provider's responsibility to define the reward function associated with a particular application/session. In our study the reward value for each request type is computed as the sum of the requests that successfully complete the buy confirm state.

4 Performance Evaluations

4.1 Dataset Specification

A dataset with the specifications as shown in table 1 was used for evaluating the performance of the proposed scheme. The data set included intrusion samples from the widely used DARPA datasets, requests from bots, and requests that would fail at the last stage due to unavailability of complete data. The information used for experimenting is as follows: Protocol, Flag, No. of error fragments, Connections having "SYN" errors, Number of compromised conditions,% of connections to the same services, % of connections to the different services, Number of connections from same source host to the same destination host, Number of connections from same source service to the same destination service, Number of connections from same destination host to the same source host, Number of connections from same destination service to same source host, Destination IP address, Destination port, Source bytes, Destination bytes.

Table 1. Dataset Specification

Record type	%of records
Requests from bots	12%
DDoS Attacks	33%
Legitimate	50%(50% of buy &50% of browse)
Incomplete	5%

4.2 Testing Phase I

CAPTCHA test filtered about 10% of the requests that were generated from bots and with 8 attributes (entropy of source IP address, entropy of src_port number, entropy of destination IP address, entropy of dest_port number, entropy of packet type, occurrence rate of packet type and number of packets) about 81.9 % of the DDoS attacks were detected by our detection module that used k-NN algorithm. Performance improvement realized by implementing the k-NN classifier is shown in table 2.

Table 2. Dataset Specification

Scheme	No.of Malicious Requests / Total Number of requests	Correct Classification	Mis-classification	% of filtering
MLF with FCM	**165/500**	132	33	81.2
MLF with k-NN	**165/500**	136	29	81.9

The classification of records using FCM and the k-NN methods are shown in figure 2a and 2b respectively.

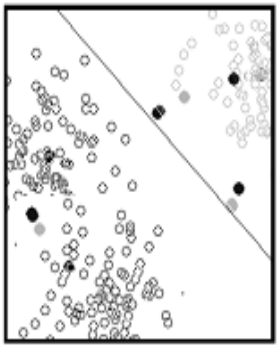

Fig 2a: FCM

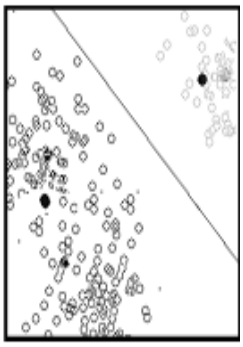

Fig: 2b k-NN

4.3 Testing Phase II

Our experimental results show that that under normal load the web server serves requests with longer sessions and at times of peak load requests with shorter session times are given priority and as the number of incoming requests increases the number of aborted sessions also increases. The web server performance with MPAC scheme is illustrated in the graphs of figure 3a and 3b.

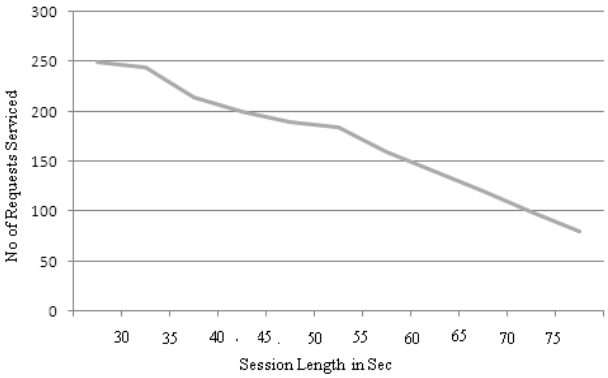

Fig 3: a. Throughput in completed sessions

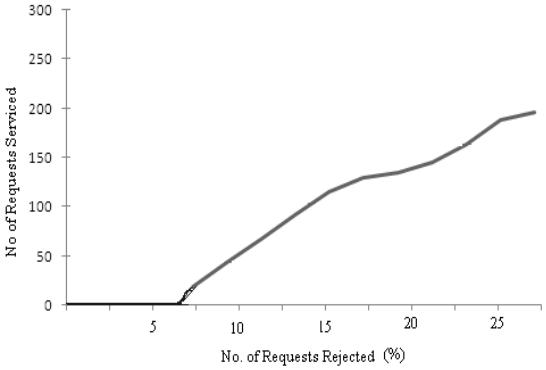

Fig 3: b. Percentage of aborted and admitted sessions

Performance of E-Commerce service with MLF and MLF integrated with MPAC is illustrated in Figure 4. Figure 4a shows the performance of the MPAC scheme compared to the Admission scheme of MLF. In the case of admission scheme used in MLF it was found that the throughput decreased, because of resource competition from browsing and ordering mix clients. With MPAC the throughput degradation is comparatively slower. The fact that the performance of MPAC scheme improves with knowledge about the customer behavioral pattern is clear from the graphs in Figure 4b and 4b. It is evident that the scheme accepts all the premium customers' session requests, and admits other customers' session requests based on the remaining of system capability. Irrespective of the load on the system, MPAC admission control can always give high priority to the session

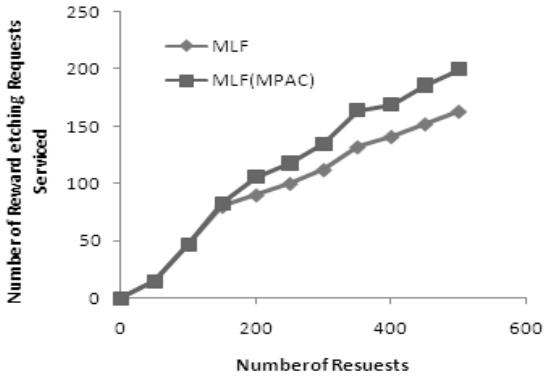

Fig 4: a. **Throughput Achieved With and without MPAC**

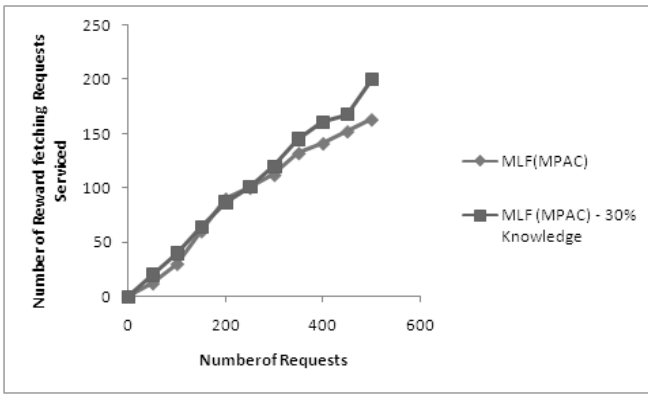

Fig 4: b. With 30% knowledge about customer behavior

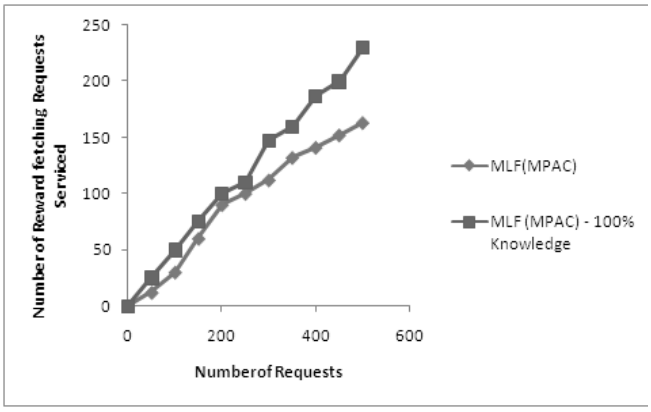

Fig 4: c. With 100% knowledge of customer behavior

requests from premium customers thereby providing the maximal profit for an E-Commerce Web site

5. Conclusions

The primary goal has been to enhance the performance of MLF scheme and enable real world applications to take advantage of this functionality. The paper presents and evaluates a strategy that improves attack detection and admission strategy to maximize the reward for having serviced. The need for prioritized service in E-Commerce sites is well-understood, as not all customers are of equal importance, and thus some should receive higher priority service over others. Allowing access to web servers through specialized access-nodes achieves drop of malicious attacks well ahead of the Web server thereby improving the performance of a small subset of high-priority requests. It is obvious from obtained results that a web server integrated with an admission control mechanism provides the required web quality of service guarantees. Our observation was that there was a remarkable improvement in the performance of the web server with the replacement of the FCM classifier by k-NN and the simple admission control policy used by MLF with an enhanced version based on customer type.

6. REFERENCES

[1] N. Harini and Dr. T.R. Padmanabhan, "A Secured-Concurrent-Available architecture for improving performance of web servers", Journal of Communications in Computer and Information , August 2012, Springer.

[2] CERT statistics. Available at:

http://www.cert.org/stats/cert_stats.html.

[3] Z. Fengxiang, Sh.ABE. A Heuristic DDoS Flooding Attack Detection Mechanism Analyses based on the Relationship between Input and Output Traffic Volumes. Computer Communications and Networks. 2007, pp. 798-802.

[4] Hoai-Vu Nguyen and Yongsun Choi. "Proactive Detection of DDoS Attacks Utilizing k-NN Classifier in an Anti-DDos Framework", International Journal of Electrical and Electronics Engineering, 2010.

[5] T. Wilson. E-Biz Bucks Lost under SSL Strain. Internet Week Online. May 20, 1999.

http://www.Internetwk.com/lead/lead052099.Html

[6] I. Schmerken, Can the Market's Systems Keep Up With Electronic Trading?, Wall Street & Technology, February 2007.

[7] Novella Bartolini, Giancarlo Bongiovanni, Simone Silvestri, " Self-* through self-learning: overload control for distributed web systems",2008

[8] "E-Commerce facts posted by ZDNet Research,

http://blogs.zdnet.com, 2007.

[9] Alexander Totok, VijayKaramcheti, "RDRP: Reward-Driven Request Prioritization for E-Commerce Web Sites", Journal of Electronic Commerce Research and Applications, March 2010