Predictive Fuzzy Request Control Mechanism in Virtualized Server Environment

San Hlaing Myint
University of Computer Studies,Yangon
shm2007@gmail.com

Abstract

In virtualized server environment, various services are revealed to high rates of incoming requests. The servers may become overloaded during temporary traffic peaks when more requests arrive than the server is considered for. Request drop rate increase more and more because of unfair overload prediction decision among admission control method. In this paper, Predictive Fuzzy Request Control mechanism (PFRCM) was developed for preventing and controlling server overloads and reducing request drop rate, and for getting response time guarantee for user satisfaction. In this work, overload prediction decision is made by considering three main parameters. The information of overload predictor support effectively for admission control (AC) process. So proposed PFRCM can manage multiple requests and make server more stable and rapid by reducing request drop rate. And then virtualized server architecture is set up in our test bed by using Xen hypervisor technology.

Keywords— virtualization, fuzzy logic, dmission control

1. Introduction

There are many challenges arising from server virtualization and the management of virtual environments to identify the solutions needed to resolve virtualization issues and automated data center management. One of the key challenges is the capability of acquiring and responding service request on demand at least request drop rate. Providing the services request automatically during appropriate response time to satisfy Service level objectives (SLOs) .One key feature of virtualized datacenter environment is to satisfy the following Quality of service (Qos); a fair chance of completion for any accepted session.

Some conventional control approaches based their request discard decisions on hard thresholds. Traditionally, server utilization or queue lengths have been the variables mostly used in admission control schemes. In this work, the response times and Transaction per second (TPS) was chosen as the control parameter since it indirectly affects system utilization and system overloading. TPS is the number complete request during an appropriate time. We refer TPS as server throughput. For virtualized servers, the main objective of the control scheme is to protect it from overload situation. When server gets overloaded, the response time for a service becomes long which affects the many users. One of the best solutions is admission control mechanism can be implemented in the servers. The admission control mechanism handles some requests with fair admission decision whenever the arriving traffic is too high and thereby maintains an acceptable load in the system.

When a server gets overloaded, the response time for a service becomes long which affects the many visitors. Longer response time not only depends on server's overload but also server throughput. Most of overload prediction method makes their prediction decision depend on only response time. In this work, proposed intelligent predictor focuses not only response time but also server throughput for decision making process. So we adjust trade-off between response time and throughput.

In this paper, we proposed a fuzzy logic request control system in virtualized server architecture in order to obtain the efficiency of the fuzzy approach which closely follows the reference utilization and lower response time and higher throughput, with less variability for all request classes considered. This was possible because the unique characteristic of the fuzzy approach allows a fine tuning of request admission and discard decisions.

Three issues are considered for proposed mechanism:

First, a hierarchical control architecture that adjusts admission control of virtualized servers is developed to achieve end-to-end mean response time targets for the multiple services that virtualized servers provide and to meet the time varying demand of client's request.

Second, a fuzzy logic based intelligent predictive model is developed that captures the health of Physical server which integrate to admission controller to make accurate admission decision by adjusting input parameters, Response Time (T_r), Service Time (T_s), Queuing Delay (T_d) and Transactions per Second (TPS) which are used real-world server logs in a trace-driven simulation.

Third, proposed PFRCM in Xen virtualized environment is validated by using RUBiS workload load generators for evaluating experimental result. Experimental results demonstrate that proposed controller make fair decision for admission of request and maintains application response time at specific target and avoiding server overloading and thrashing problems.

The rest of the paper is organized as follows. Section 2 presented related work. In Section 3, motivation of admission control was described. Section 4 presents proposed methodology. The fuzzy steps of proposed PFRCM are explained clearly in section 5. An experimental test bed was represented in Section 6. Section 7 describes numerical evaluation. Finally, we summarize our conclusions.

2. Related Work

Prepare In wide-ranging, Admission control (AC) is related to problems in computer networks. One can realize the admission process as the obligation by the service provider to supply the service requested by the customer. In networks with multimedia applications, acceptance of a request shall consider a set of parameters related to QoS, according to principles determined by an appropriate policy. In this circumstance, the fuzzy logic has emerged as a powerful tool for shaping the rules for acceptance or refusal of a request.

Author proposed a dynamic session management based on reinforcement learning for utilizing machine resource and avoiding thrashing in [1].Reference [2] proposed a method for admission control to control overload in web servers by utilizing neural network. The most of learning method requires a lot of learning time and learning agent estimated the amount of used resource only depend on response time.

The incoming request managing defines mechanisms for resource allocation to regulate the network in agreement with the variation of traffic. In [3], a controller based on fuzzy logic reconfigures the nodes of a field according to the incoming traffic. Fuzzy controllers in non-linear systems add one more challenge: adjust the parameters of the controller using heuristic optimizers.

In [4], the use of fuzzy controllers is often vindicated by the non-linearity or lack of a precise mathematical model to address the changes in system state. In general, the rules defining the quality policy of the service in these environments involve imprecise parameters. In such state of affairs, fuzzy logic eases the demonstration of QoS policy rules.

In a virtualized server environment, a conventional proportional-integral controller is overcome by a fuzzy controller [7]. Another likelihood investigated recently performs admission control considering the forecast of work-load caused by customers [5]: a Support Vector Machine (SVM) was used to predict a time series representing the workload which the web server will be submitted to.

Reference [8] proposed the disadvantage of the classical PI controller is that the system model, which is obtained by system identification, mismatches the real system and inevitably

degrades the performance of the web system. In contrast with classical PI controllers, fuzzy controllers are nonlinear and therefore independent of the accurate model of the plant, i.e. the controlled system. Hence, fuzzy controllers seem to be very suitable for web servers.

Most of admission control mechanism considered only response time as control parameter and they can provide for single service server like web server. Only response time is not sufficient to estimate about overload because overload can be caused by other factors like bandwidth, processing time, queuing delay and arrival rate etc. Therefore, their estimation results are very general and these methods cannot control request for multiple virtualized server.

3. Admission Control

For When Internet service experiences transient high user demand, restricting its availability to users is necessary to avoid complete service breakdown. Admission control protects Internet services by preventing resource overload through policing and selective acceptance of incoming traffic. Admission control employed at the request level can certainly protect the service from overload. However it might result in resource wastage in the form of rejected request. To maximize throughput of a multi-tier Internet service, admission control ought to be employed at request level. The service quality and responsiveness of a request based system are typically measured by the absolute performance metrics, response time and queuing delay. However, these metrics do not take into consideration the varying demands posed by the different requests. It is known that clients are more likely to anticipate short delays for "small" requests and more willing to tolerate long delays for "large" requests [6]. On the other hand, it is not sufficient to estimate about overload because overload can be caused by other factors like bandwidth, processing time, queuing delay and arrival rate etc Because these metric directly translates to user-perceived relative performance and system load, it has been accepted as an important performance metric on servers.

4. Proposed Methdology

The proposed architecture of PFRCM is divided into two main categories: an Admission Control (AC) module and fuzzy overload predictor (FOP), according to Figure (1). The innovation of components is the use of fuzzy logic. Overload predictor estimates the server overload from the response time for a service request, Service Time, Queuing Delay and transaction per second (TPS), and makes an accurate prediction result of server overloading. The Admission Control (AC) module handles the acceptance of new requests by the server taking into account current service policies and system workload information according to overload predictor. In case of system overload, a request may be either rejected (Dropping) until server is recovered. Virtualized server Model is considered in order to satisfy the following assumptions.

- V_i denotes Virtualized server that provides serve i.
- The number of virtualized servers (V_i) on a physical server (P) is M.
- Virtualized server V_i provides Service i ($1 \leq i \leq M$)
- The maximum number of requests that can accept for M is X_i.
- The number of accepted requests for Vi at time t is $L_i(t)$($Li(t) \leq X_i$)
- The no of request that can access the V_i is R_i.

A testbed are created according to this assumption, so proposed method can be implemented easily. In this system, web services and database services are setup in virtualized servers for providing difference services.

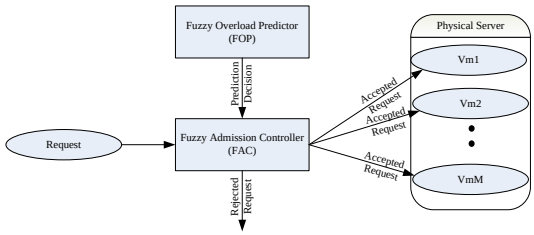

Figure 1. Overview of predictive fuzzy request control mechanism

5. Fuzzy Logic Based Request Control Mechanism

Number In many of the AC algorithms, the computational complexity becomes very high, mainly due to the number of states that describe the system and it is very hard to allocate or to decide the accurate values for the thresholds.
Fuzzy logic can be considered that can facilitate to overcome these problems. The use of fuzzy logic can reduce the computational complexity compared to Markov models. The threshold-based AC algorithms can be easily extended through the framework of fuzzy logic. Another significant advantage is the ability of fuzzy logic to accept rules expressed in a natural language and to combine the rules with facts through the fuzzy inference, obtaining the conclusions. In web server environment, fuzzy logic is used for admission control in [9], but only in relation with fixed interval (e.g. token bucket algorithm).

5.1. Fuzzy Inference

The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations. The operations on fuzzy sets are differenced than the operations on non-fuzzy sets. Let $\mu_A$and $\mu_B$are the membership functions for fuzzy sets A and B.
Given a fact A' and a rule RA→B, fuzzy inference means the composition A'○ RA→B in order to obtain the conclusion B'= A'○ RA→B. In this work, fuzzy inference the computation procedure is used, like in [23]. According to this procedure, the membership function of B' is:

$$\mu_{B'}(y)=max_{x\in U}\ min\ (\mu_{A'}(x),\mu_{R:A\to B'}(x,y)) \quad (1)$$

where:

$$\mu_{R:A\to B'}(x,y)) = \min_{A,B}(\mu(x),\mu(y)) \quad (2)$$

After transformation (1) results in:

$$\mu_{B'}(y) = \min_{B}(a,\mu(y)) \quad (3)$$

Where $a = \max_{x\in U} \min\ (\mu_{A'}(x), \mu_A(x))$

In equation (1) – (3), we have used the following notations;μ_A (X),$\mu_{A'}$ (X),μ_B (X),$\mu_{B'}$ (X) are the membership functions of the fuzzy sets A (a term in premise), A' (the fact), B (a term in conclusion), B' (the consequent of the rule) , defined over the universe of discourse of the fuzzy sets, and taking values in the closed real interval [0,1]. The universe of discourse, or the domain, for the sets A and A' (the linguistic variables in premise and the facts) is U, and for the fuzzy sets B and B' is V. If the premises are composed, the degrees of activation of each premise (a in (3)) are combined using the corresponding logical operators (AND, OR, NOT). We use only the AND operator, which is implemented by minimum in fuzzy logic.
When more than one rule is active, the consequents of all active rules are combined through the union operator, which is implemented as a maximum between the membership functions of the partial conclusions. Most often, the result of the fuzzy inference has to be a crisp value, obtained by an averaging procedure applied on the partial conclusions, process that is called defuzzification. A widely used defuzzification method is the center of gravity.

5.2. Fuzzy Logic controller (FLC)

The fuzzy inference is performed by a fuzzy logic controller (FLC) that can be implemented either hardware or software. A linguistic term is a fuzzy set, which is a set characterized by a membership function that can take values in the closed interval [0,1], while the membership function for a non-fuzzy (crisp) set can take only the discrete values 1 or 0 (an element either belongs to the set, or it doesn't belong to that

set). In many cases, the "shapes" chosen for the membership functions are linear triangles or trapezes shapes are adopted for the fuzzy terms in premises. They are represented in Figure 2 Figure 3 and Figure 4, where a1, a2,b1, b2 are parameters that can be adjusted. The domains U and V of the linguistic variables in premises and conclusion respectively are mapped in fuzzy logic predictor and controller to the closed interval [0, m].

In fuzzy logic system, a rule base is constructed to control the output variable. A fuzzy rule is a simple IF-THEN rule with a condition and a conclusion. In this work, the set of rules used are in the form: "if response time is high and Transactions per Second (TPS) is low then prediction decision states that server is in overload state", where the part between if and then is the (composed) premise of the rule, and the rule conclusion follows after then. In this example, the linguistic variables are "Response Time (T_r)", "Service Time(T_s)", "Queuing Delay(T_d)" ,"Transactions per Second (TPS)" and "prediction decision", each linguistic variable having a number of terms. For example, the terms three parameters and for Transactions per Second (TPS) are high (H), medium (M) and low (L). The terms for the linguistic variable "prediction decision" in conclusion are overloaded state (OS), normal state (NS) and idle state (IS). For admission decision, the terms for physical server and virtualized web server(Vi) are overload(O), normal (N) and idle (I).In conclusion, the linguistic variable are Strongly Reject(SR),Reject(R),Strongly Accept(SA), and Accept(A). The rules are completely described in Table 1,2,3 and Table 4:

Table 1. The Fuzzy Rules for Predictor

		Transactions Per Second (TPS)		
		L	M	H
Response Time(T_r)	L	NS	IS	IS
	M	OS	NS	IS
	H	OS	OS	NS

Table 2. The Fuzzy Rules for Predictor

		Transactions Per Second (TPS)		
		L	M	H
Service Time(T_s)	L	NS	IS	IS
	M	OS	NS	IS
	H	OS	OS	NS

Table 3. The Fuzzy Rules for Predictor

		Transactions Per Second (TPS)		
		L	M	H
Queuing Delay(T_d)	L	NS	IS	IS
	M	OS	NS	IS
	H	OS	OS	NS

Table 4. The Fuzzy Admission Controller

		Virtualized Server (Vi)		
		O	N	I
Physical server(P)	O	SR	R	A
	N	R	A	SA
	I	A	SA	SA

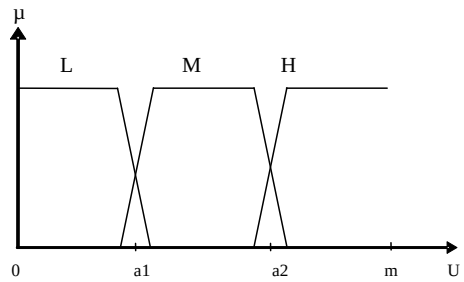

Figure 2.The fuzzy terms in premises (response time)

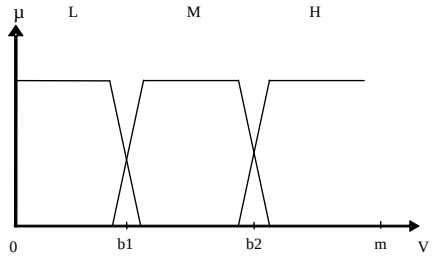

Figure 3. fuzzy terms in premises (transaction per second)

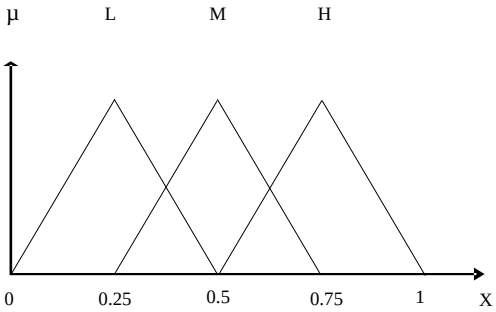

Figure 4.The fuzzy terms in conclution

6. Experimental Testbed

In order for testing proposed method, it is necessary to setup hypervisor virtualization technology like Xen and Apache which are widely used to build multiple virtualized servers on physical machine. In this work, Xen hypervisor virtualization technology is used for implementing proposed method. We implement this C program into domain 0 which is for request management see in Figure(5).when a new request is arrive at physical server, it executes a command according to optimal action that is derive with proposed PFRCM. Thus proposed method can be implemented into virtualized server environment easily.

However, to measure server performance and evaluate proposed method, it is necessary to run RUBiS workload generator on client machines. RUBiS clients will produce request so that push RUBiS component to their limited, so that resource like CPU usage and network throughout get high value.

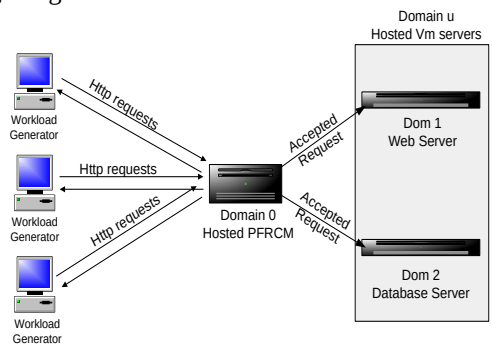

Figure 5. Implementation of proposed method (PFRCM)

7. Numerical Evaluation

7.1. Numerical calculation of Overload Prediction

In this section, the situation that can occur system overload is described by using mathematical calculation. The following assumptions are defined for calculation. RT_i, ST_i and RT_i denotes response time, service time and queuing delay time of virtualized server i that provide service i ($1 \le i \le M$).T_r ,T_s and T_d are denoted for smallest value of RT_i, ST_i and RT_i. In this work, the number of virtualized server is M=2 and this servers provide service i (i =1,2). System overload is calculated by calculating error value;

$$E_r(t) = \sum_{I=1}^{M} (RT_i(t) - T_r) \quad (4)$$

$$E_s(t) = \sum_{I=1}^{M} (ST_i(t) - T_s) \quad (5)$$

$$E_d(t) = \sum_{I=1}^{M} (DT_i(t) - T_d) \quad (6)$$

Where;

E_r = error value between actual response time and smallest response time

E_s = error value between actual service time and smallest service time

E_t = error value between actual queuing delay time and smallest queuing delay time

According to numerical calculation;
when $E_r(t) \geq 0.2$, $E_s(t) \geq 1.0$ and $E_d(t) \geq 0.2$, system overload is occur. The greater error value, the more overload causes. When the error value is zero, the system gets the best performance.

7.2. Impact of Proposed Predictive Method on Request Drop Rate

In this section, effectiveness of proposed method on the request drop rate is investigated. In figure (6), result of proposed method is denoted as "with PFRCM" but the result of the existing method is denoted as "without PFRCM". The figure shows the request drop rate against the total number of requests for physical server.

According to the figure, request drop rate increased when the total number of arriving request is much smaller about 30000 per second in existing method. On the other hand, the proposed method can reduce the request drop rate until the total number of arriving request is up to about 50000 per second. In proposed method, although requests something drops under 50000 of arriving request per second, it is not significant. Proposed PFRCM can reduce request drop rate effectively.

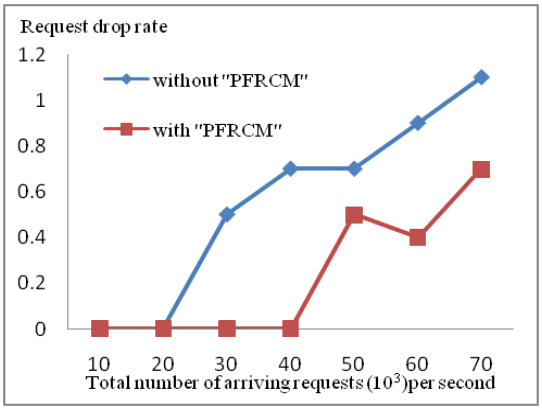

Figure (6).Request drop rate Vs number of arriving requests.

8. Conclusions

This paper proposed a framework for controlling multiple requests in virtualized server environment by using fuzzy logic theory. Fuzzy-logic based predictive request control mechanism (PFRCM) is implemented in virtualized server environment in order to obtain lower response time and higher throughput, with less variability for all request classes considered. In this work, proposed intelligent predictor adjusts between response time and server throughput for making prediction process accurately, so fuzzy controller make admission decision fairly among multiple services request. The obtained results show that predictive admission control with optimal configuration of the considered of virtualized servers effectively reduced request drop rate. In the future work, we will consider other criteria like bandwidth, processing time for predicting server health and we will implement other intelligent machine learning method to optimize server performance. To simulate the workload, we'll use real-world server logs in a trace-driven simulation.

References

[1] Albert Hidalgo Barea, "Analysis and evaluation of high performace web servers",july 13 th 2011,Universitat Politecnica De Catalunya.

[2] Kimihiro Mizutani, Izumi Koyanagi, Takuji Tachibana," Dynamic session management based on reinforcement learning in virtual server environment," in proc.of IMECS 2010 Vol II, March 17-19,2010,Hong Kong

[3] H.-J. Zimmermann, "Fuzzy sets theory – and its applications. Second, revised edition," Kluwer Academic Publishers, 1991.

[4] M.J. Patyra, "Design Considerations of Digital Fuzzy Logic Controllers," in "Fuzzy Logic - Implementation and Applications,"pp 143-175, edited by M.J. Patyra and D.M.

[5] Ka Ho Chan, Xiaowen Chu," Design of a Fuzzy PI controller to Guarantee Proportional Delay Differentiation on Web ervers," COMP-06-001:October 16, 2006.

[6] M. Harchol-Balter. Task assignment with unknown duration. Journal of ACM, 29(2):260–288, 2002.

[7] V. Cardellini, E. Casalicchio, and M. Colajanni, ”A performance study of distributed architectures for the quality of web services,” In IEEE Int. Conf. on System Sciences (HICSS ’01), pages 3551–60, jan 2001.
[8] M. P. Fernandez, A. C. P. Pedroza, and J. F. de Rezende.Otimizac ¸”˜ ao de controlador fuzzy para provisionamento de recursos em ambiente diffserv atrav´ es de algoritmo gen´ etico,”In SBRC 2002, page 16, 2002.
[9] V. Cardellini, E. Casalicchio, M. Colajanni, and M. MamBelli, “Web switch support for differentiated services,” ACMPerformance Evaluation Review, 29(2):14–19, sep 2001.
[10] V. Cardellini, E. Casalicchio, and M. Colajanni,” A performance study of distributed architectures for the quality of web services,” In IEEE Int. Conf. on System Sciences (HICSS ’01), pages 3551–60, jan 2001.
[11] Y.-C. Ko, S.-K. Park, C.-Y. Chun, H.-W. Lee and C.-H. Cho, “An adaptive QoS provisioning distributed call admission control using fuzzy logic control,” in Proc. of IEEE ICC 2001, vol. 2, pp. 356-360, Helsinki, Finland, 2001.
[12] Y.Wei, C. Lin, X. Chu, and T. Voigt, “Fuzzy control for guaranteeing absolute delays in web servers,” High Performance Computing and Networking 2006, 4(5/6):338–346, 2006.
[13] RUBiS,http://rubis.ow2.org/
[14] ApacheSoftware Foundation.http://www.apache.org
[15] May 2011 Web server survey by Netcraft, http://news.netcraft.com