Appl. Math. Inf. Sci. 6 No. 2S pp. 471S-477S (2012)

Applied Mathematics & Information Sciences
An International Journal

@ 2012 NSP
Natural Sciences Publishing Cor.

Efficient Content Placement on Multimedia CDN Using Fuzzy Decision Algorithm

Jian-Bo Chen
Department of Information and Telecommunications Engineering, Ming Chuan University, 333, Taoyuan, Taiwan
Corresponding author: Email: jbchen@mail.mcu.edu.tw

Received June 22, 2010; Revised March 21, 2011; Accepted 11 June 2011
Published online: 1 January 2012

Abstract: Multimedia Content Delivery Networks (CDN) have been employed on the Internet as a way to improve performance and reliability. In CDN architecture, the content is duplicated from the origin server to replica servers at the edge of the Internet to improve the performance and minimize the use of network bandwidth. Efficiently placing the multimedia contents in CDN is a challenging problem. The bandwidth availability, content hit rate, and storage availability are three factors that can be used to determine the placement of multimedia contents. In this paper, a fuzzy decision algorithm is adopted to solve this issue. With the proposed fuzzy decision algorithm, three input parameters are used. The fuzzy decision algorithm consists of three steps. The *fuzzification* step obtains the membership function values from the three input parameters. The *rule evaluation* step uses these membership function values to obtain the fuzzy decision values. And the *defuzzification* step obtains the final crisp value based on the fuzzy decision values. This crisp value is used to determine the placement of multimedia contents. The simulation results show that the fuzzy decision algorithm can achieve a higher performance than other algorithms.

Keywords: CDN, Content Placement, Fuzzy Decision Algorithm

1 Introduction

The CDN (Content Delivery Network) is an architecture that deploys a set of replica servers distributed throughout the Internet for better performance and availability than a centralized architecture.[1-4] Akamai [5], Exodus [6], and Digital Island [7] are examples of commercial content providers that use CDN architecture.

There is a variety of content that can be distributed throughout the Internet. For example, streaming multimedia, hierarchical web applications, advertisements, general web content, and mass storage or software are popular contents that will be distributed in different replica servers to improve the performance. In additional, there are some replica placement algorithms that are used to determine which contents will be duplicated to which replica server. These algorithms include the tree-based algorithm, greedy algorithm, random algorithm, and hot spot algorithm.[8] This paper focused on the content placement in multimedia streaming CDN environment.[9]

In [8], the authors indicate that hot spot algorithm is the best choice for this kind of content since latency is minimized. Although hot spot algorithm is the best choice for multimedia streaming contents, this paper will propose another intelligent algorithm, fuzzy decision algorithm[10-12], to improve performance. The fuzzy decision algorithm consists of three steps: *fuzzification*, *rule evaluation*, and *defuzzification*. The follows are the input parameters for the fuzzy decision algorithm: bandwidth availability, content hit rate, and storage availability. In the *fuzzification* step, the membership function values will be obtained through these three input parameters. Based on the *rule evaluation* step, four fuzzy decision values can

be obtained from fuzzy rule bases. Then the final crisp value can be obtained through the *defuzzuifcation* step. The final crisp value can be used to determine which content is suitable to be replicated to which servers. The simulation results show that the proposed fuzzy decision algorithm can achieve higher performance than other algorithms.

This paper is structured as follows: Section 2 discusses previous works. Section 3 discusses the fuzzy decision algorithm processes. Section 4 gives the simulation results. Section 5 offers conclusions.

2 Previous Works

In this section, some algorithms that can be used in replica placement are discussed. These algorithms include the tree-based algorithm, greedy algorithm, random algorithm, and hot spot algorithm.[13]

2.1 Tree-based Algorithm

In [14], the authors propose an optimal placement of web proxies in the Internet. The authors argue that the placement of web proxies is critical to performance and they further investigate the optimal placement policy of web proxies for a target web server in the Internet. The objective is to optimize a given performance measure for the target web server subject to system resources and traffic patterns. They are interested in finding the optimal placement of multiple web proxies in a number of potential sites under a given traffic pattern. This approach is based on the assumption that the network structure is a tree. The origin server is the root of the tree. The clients make requests from the closest replica server in their path toward the root. This algorithm was originally designed for web proxy cache placement, but it is also applicable for content placement.

2.2 Greedy Algorithm

The basic idea of the greedy algorithm is that it takes into account the existing information from CDN, such as workload patterns and the network topology. It needs to choose *M* replicas among *N* potential sites; it chooses one replica at a time. The first iteration evaluates each of the *N* potential sites individually to determine their suitability for hosting a replica. It computes the cost associated with each site under the assumption that access from all clients converge at that site, and then picks the site that yields the lowest cost. The second iteration searches for a second replica site which yields the lowest cost. In general, in computing the cost, it assumes that clients direct their access to the nearest replica which can be reached with the lowest cost. The iteration will stop when the *M* replicas have been chosen.

2.3 Random Algorithm

The random algorithm chooses *M* sites from *N* places randomly. To improve performance, the algorithm runs several times and gets the best result which yields the lowest cost.

2.4 Hot Spot Algorithm

The hot spot algorithm attempts to place replica contents near the clients generating the greatest load. It sorts the *N* potential sites according to the amount of traffic. Then it places the replicas at the top *M* sites that generate the largest amount of traffic.

3 Fuzzy Decision Algorithm

In this section, the fuzzy decision algorithm will be introduced. This algorithm includes three steps: *fuzzification*, *rule evaluation*, and *defuzzification* as shown in Fig. 1.

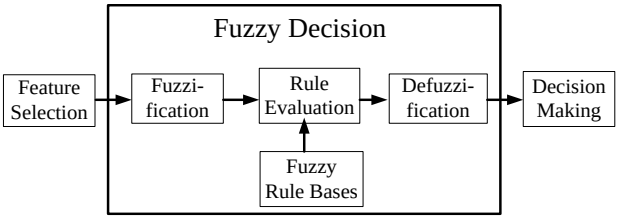

Figure 1. The three steps for the fuzzy decision algorithm

In this paper, three features are selected as the input parameters for the fuzzy decision algorithm. The membership functions are defined as high degree, medium degree, and low degree. After the *fuzzification* step, nine membership function values are obtained based on the features. Therefore, these nine membership function values yield twenty-seven combinations, in addition to four fuzzy decision values that can be obtained from referencing a table of these twenty-seven combinations and fuzzy rule bases. After the *defuzzification* step, the final crisp value can be obtained according to these four fuzzy decision values. The fuzzy process is shown in Fig. 2 and the detail will be described in this section.

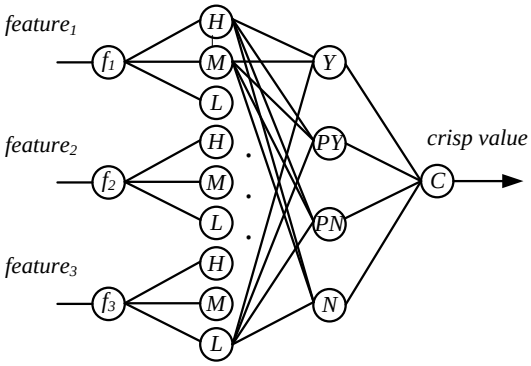

Figure 2. The fuzzy process

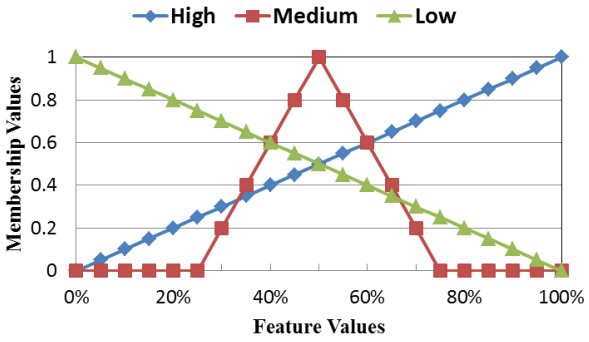

Figure 3. Membership functions

3.1 Fuzzification

Before the *fuzzification* step, three features must be defined and normalized. The first feature, *bandwidth availability*, named as f_1 is defined as

$$f_1 = (B_T - B_C) / B_T \tag{3.1}$$

where B_T is the total bandwidth and B_C is the current bandwidth used. The second feature, *content hit rate*, named as f_2 is defined as

$$f_2 = H_C / H_T \tag{3.2}$$

where H_T is the total hit count and H_C is the hit count for that content. The third feature, *storage availability*, named as f_3 is defined as

$$f_3 = (S_T - S_C) / S_T \tag{3.3}$$

where S_T is the total storage and S_C is current storage used.

In this proposed fuzzy decision algorithm, three membership functions for each feature are defined. The *High*, *Medium*, and *Low* membership functions are defined as the high degree, medium degree, and low degree membership functions, respectively. The definitions of these membership functions are shown Fig. 3.

The definition of *High* membership function is

$$f(x) = x, \ for\ 0 \le x \le 1 \tag{3.4}$$

The definition of *Medium* membership function is

$$f(x) = \begin{cases} 0, & for\ 0 < x \le 0.25 \\ \frac{x}{0.25} - 1, & for\ 0.25 < x \le 0.5 \\ 3 - \frac{x}{0.25}, & for\ 0.5 < x \le 0.75 \\ 0, & for\ 0.75 < x \le 1 \end{cases} \tag{3.5}$$

The definition of *Low* membership function is

$$f(x) = 1 - x, \ for\ 0 \le x \le 1 \tag{3.6}$$

Based on these definitions, the three membership functions for *bandwidth availability*, *content hit rate*, and *storage availability* can be obtained as shown in Fig. 4 to 6. There are total nine *membership function values (MFV)* as shown in Table. 1. Where *HBW* is the abbreviation for *high bandwidth availability*, etc.

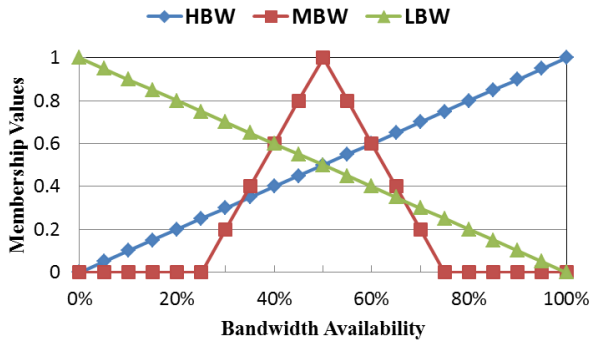

Figure 4. Membership function for bandwidth availability

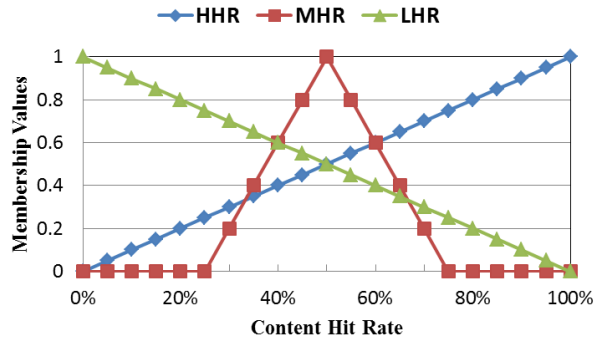

Figure 5. Membership function for content hit rate

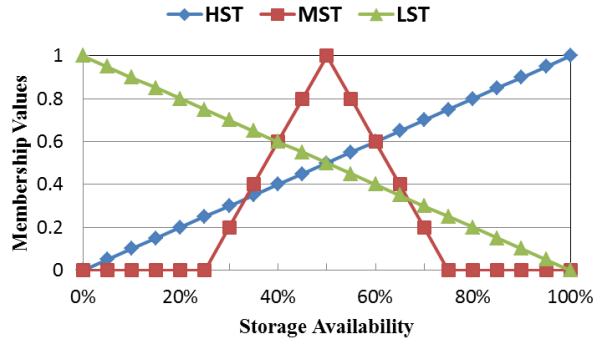

Figure 6. Membership function for storage availability

Table 1. The nine membership function values

Abbreviation	*Meaning*
HBW	*High Bandwidth*
HHR	*High Hit Rate*
HST	*High Storage*
MBW	*Medium Bandwidth*
MHR	*Medium Hit Rate*
MST	*Medium Storage*
LBW	*Low Bandwidth*
LHR	*Low Hit Rate*
LST	*Low Storage*

For example, if the bandwidth availability is 30%, the content hit rate is 60%, and the storage availability is 55%, then after *fuzzification*, nine membership function values are obtained as shown in Fig. 7.

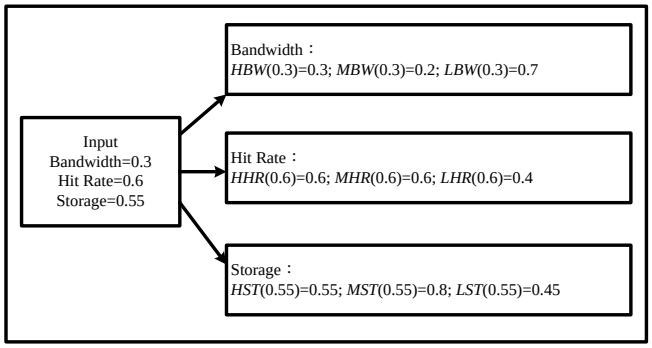

Figure 7. Example of fuzzification

3.2 Rule Evaluation

When the nine *MFV*s are obtained, the next step is *rule evaluation*. There are 3^3=27 combinations by (*HBW*, *MBW*, *LBW*), (*HHR*, *MHR*, *LHR*), (*HST*, *MST*, *LST*). For each combination, the rule bases define the fuzzy decision as *Y*, *PY*, *PN*, and *N*, which represents the fuzzy decision is *Yes*, *Probably Yes*, *Probably No*, and *No*, respectively. The fuzzy rule bases are shown in Table 2.

Table 2. The rule evaluation bases

Bandwidth	*Hit Rate*	*Storage*	*Rule Evaluation*
HBW	*HHR*	*HST*	*Y*
HBW	*HHR*	*MST*	*Y*
HBW	*HHR*	*LST*	*PY*
HBW	*MHR*	*HST*	*Y*
HBW	*MHR*	*MST*	*Y*
HBW	*MHR*	*LST*	*PY*
HBW	*LHR*	*HST*	*PY*
HBW	*LHR*	*MST*	*PY*
HBW	*LHR*	*LST*	*PN*
MBW	*HHR*	*HST*	*PY*
MBW	*HHR*	*MST*	*PY*
MBW	*HHR*	*LST*	*PY*
MBW	*MHR*	*HST*	*PY*
MBW	*MHR*	*MST*	*PN*

MBW	*MHR*	*LST*	*PN*
MBW	*LHR*	*HST*	*PY*
MBW	*LHR*	*MST*	*PN*
MBW	*LHR*	*LST*	*PN*
LBW	*HHR*	*HST*	*PY*
LBW	*HHR*	*MST*	*PN*
LBW	*HHR*	*LST*	*PN*
LBW	*MHR*	*HST*	*PN*
LBW	*MHR*	*MST*	*N*
LBW	*MHR*	*LST*	*N*
LBW	*LHR*	*HST*	*PN*
LBW	*LHR*	*MST*	*N*
LBW	*LHR*	*LST*	*N*

Each rule evaluation value (*REV*) is calculated by the average of the three *MFV*s. The definition for *REV* is shown in Eq. 3.7, where *i* from 1 to 27 represent the twenty-seven combinations. And the example of the twenty-seven values is shown in Table 3.

$$REV_i = average(HBW/MBW/LBW,\ HHR/MHR/LHR,\ HST/MST/LST) \quad (3.7)$$

Table 3. Example of rule evaluation

Rules	*Rule Evaluation Values*
Rule(HBW, HHR, HST)=Y	*average(HBW=0.3, HHR=0.6, HST=0.55)=0.48*
Rule(HBW, HHR, MST)=Y	*average((HBW=0.3, HHR=0.6, MST=0.8)=0.57*
Rule(HBW, HHR, LST)=PY	*average((HBW=0.3, HHR=0.6, LST=0.45)=0.45*
Rule(HBW, MHR, HST)=Y	*average((HBW=0.3, MHR=0.6, HST=0.55)=0.48*
Rule(HBW, MHR, MST)=PY	*average(HBW=0.3, MHR=0.6, MST=0.8)=0.57*
Rule(HBW, MHR, LST)=PY	*average(HBW=0.3, MHR=0.6, LST=0.45)=0.45*
Rule(HBW, LHR, HST)=PY	*average(HBW=0.3, LHR=0.4, HST=0.55)=0.42*
Rule(HBW, LHR, MST)=PY	*average(HBW=0.3, LHR=0.4, MST=0.8)=0.5*
Rule(HBW, LHR, LST)=PN	*average(HBW=0.3, LHR=0.4, LST=0.45)=0.38*
Rule(MBW, HHR, HST)=PY	*average(MBW=0.2, HHR=0.6, HST=0.55)=0.45*
Rule(MBW, HHR, MST)=PY	*average(MBW=0.2, HHR=0.6, MST=0.8)=0.53*
Rule(MBW, HHR, LST)=PY	*average(MBW=0.2, HHR=0.6, LST=0.45)=0.42*
Rule(MBW, MHR, HST)=PY	*average(MBW=0.2, MHR=0.6, HST=0.55)=0.45*
Rule(MBW, MHR, MST)=PN	*average(MBW=0.2, MHR=0.6, MST=0.8)=0.53*
Rule(MBW, MHR, LST)=PN	*average(MBW=0.2, MHR=0.6, LST=0.45)=0.42*
Rule(MBW, LHR, HST)=PY	*average(MBW=0.2, LHR=0.4, HST=0.55)=0.38*
Rule(MBW, LHR, MST)=PN	*average(MBW=0.2, LHR=0.4, MST=0.8)=0.47*
Rule(MBW, LHR, LST)=PN	*average(MBW=0.2, LHR=0.4, LST=0.45)=0.35*
Rule(LBW, HHR, HST)=PY	*average(LBW=0.7, HHR=0.6, HST=0.55)=0.62*
Rule(LBW, HHR, MST)=PN	*average(LBW=0.7, HHR=0.6, MST=0.8)=0.7*
Rule(LBW, HHR, LST)=PN	*average(LBW=0.7, HHR=0.6, LST=0.45)=0.58*
Rule(LBW, MHR, HST)=PN	*average(LBW=0.7, MHR=0.6, HST=0.55)=0.62*
Rule(LBW, MHR, MST)=N	*average(LBW=0.7, MHR=0.6, MST=0.8)=0.7*
Rule(LBW, MHR, LST)=N	*average(LBW=0.7, MHR=0.6, LST=0.45)=0.58*
Rule(LBW, LHR, HST)=PN	*average(LBW=0.7, LHR=0.4, HST=0.55)=0.55*
Rule(LBW, LHR, MST)=N	*average(LBW=0.7, LHR=0.4, MST=0.8)=0.63*
Rule(LBW, LHR, LST)=N	*average(LBW=0.7, LHR=0.4, LST=0.45)=0.52*

The twenty-seven *REV*s belong to four groups (*Y*, *PY*, *PN*, *N*). The algorithm takes the maximum value for each group, and finally, four *Fuzzy Decision Values* (*FDV*) can be obtained: *FDV(Y)*, *FDV(PY)*, *FDV(PN)*, and *FDV(N)*. Equation 3.8 to 3.11 show the formulas of how to obtain the *FDV*s, where REV_Y represents the rule evaluation value in rule bases defined as *Y*, and *FDV(Y)* represents the fuzzy decision value for *Y*, etc.

$$FDV(Y) = max\{REV_{Y1}, REV_{Y2}, \ldots\} \quad (3.8)$$

$$FDV(PY) = max\{REV_{PY1}, REV_{PY2}, \ldots\} \quad (3.9)$$

$$FDV(PN) = max\{REV_{PN1}, REV_{PN2}, \ldots\} \quad (3.10)$$

$$FDV(N) = max\{REV_{N1}, REV_{N2}, \ldots\} \quad (3.11)$$

Following the above example, FDV(Y)= max{0.48,0.57,0.48}=0.57, FDV(PY)= max {0.45,0.57,0.45,0.42,0.5,0.45,0.53,0.42,0.45,0.38,0.62}=0.62, FDV(PN)= max{0.38,0.53, 0.42,0.47,0.35,0.7,0.58,0.62,0.55}=0.7, and FDV(N)= max{0.7,0.58,0.63,0.52}=0.7.

3.3 Defuzzification

In the *defuzzification* step, a set of weighted values are assigned to the four *FDV*s. Therefore, the *Crisp Value* (*CV*) can be determined based on these weighted values and the *FDV*s. The *CV* is shown in Eq. 3.12. where *FDV(i)* is the value of *FDV(Y)*, *FDV(PY)*, *FDV(PN)*, and *FDV(N)* and *w(i)* is the weighted value of the four *FDV*s.

$$CV = \frac{\sum_i FDV(i) \times w(i)}{\sum_i w(i)} \quad (3.12)$$

An example of weighted values is shown in Table 4. Based on an example of weighted values, the *CV* can be obtained whereby $CV = \frac{0.57\times0.4+0.62\times0.3+0.7\times0.2+0.7\times0.1}{0.4+0.3+0.2+0.1} = 0.624$. The content placement algorithm can use the *CV*s to determine which content is suitable to replicate to which server.

Table 4. Example of weighted values for each *FDV*

FD	*Weighted Value*
Y	0.4
PY	0.3
PN	0.2
N	0.1

4 Simulation Results

In this section, the average drop rate will be compared for various placement algorithms. In this simulation, each placement algorithm is executed in hundreds of simulation runs and examines the drop rate of the algorithm across all simulation runs. The simulation is based on the network topology which contains 300 replica servers. Whether the content placed in a specific replica server is determined by the various placement algorithms. It is also assume that the client request per second is from 200 to 1000, and the requested contents are random. If the bandwidth between client and replica server is not sufficient for this multimedia streaming content, then the request will be drop. The simulation result is shown in Fig. 8. As the author in [8] described, the hot spot algorithm is suitable for multimedia streaming content. But the fuzzy decision algorithm proposed here offer better results than the hot spot algorithm.

Figure 8. Simulation results

5 Conclusions

In this paper, the fuzzy decision algorithm was proposed to solve the problem of content placement on multimedia CDN environments. Although some algorithms have already been proposed to solve this problem, the fuzzy decision algorithm can offer better results. In the fuzzy decision algorithm, *fuzzification*, *rule evaluation*, and *defuzzification* are three steps to decide content placement. The *fuzzification* step obtains the membership function values for each input features. The *rule evaluation* step uses these membership function values and the rule bases to obtains the fuzzy decision values. Finally, the *defuzzification* step obtains the crisp value from the fuzzy decision values. The crisp value is used to determine the placement of multimedia contents. The simulation results show that the drop rate of proposed fuzzy decision algorithm is lower than the other algorithms.

References

[1] Yun Bai, Bo Jia, Jixiang Zhang, and Qiangguo Pu, An Efficient Load Balancing Technology in CDN, *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, (2009), 510-514.

[2] Moo-Kyung Sung and Chi-Moon Han, A Study on Architecture of CDN (Content Delivering Network) with Content Re-distribution Function, *International Conference on Advanced Communication Technology*, (2009), 772-777.

[3] S. Manfredi, F. Oliviero,and S. P. Romano, Distributed Management for Load Balancing in Content Delivery Networks, *GLOBECOM Workshops*, (2011), 579-583.

[4] Ao-Jan Su, D. R. Choffnes, A. Kuzmanovic, and F.E. Bustamante, Drafting Behind Akamai: Inferring Network Conditions Based on CDN Redirections, *IEEE Transaction on Networking*, Vol.17, No.6, (2009), 1752-

1765.

[5] Akamai, http://www.akamai.com.

[6] Exodus, http://www.exodus.com.

[7] Digital Island, http://www.digitalisland.com.

[8] M.H Al-Shayeji, S. Rajesh, M. Alsarraf, and R. Alsuwaid, A Comparative Study on Replica Placement Algorithms for Content Delivery Networks, *Second International Conference on Advances Computing, Control and Telecommunication Technologies*, (2010), 140-142.

[9] G. Fortino, W. Russo, C. Mastroianni, C.E. Palau and M. Esteve, CDN-Supported Collaborative Media Streaming Control. *IEEE Transaction on Multimedia*, Vol.14, No.2, (2007), 60-71.

[10] Jin-Long Wang and Chen-Wen Chen, Fuzzy Logic Based Mobility Management for 4G Heterogeneous Networks, *Knowledge-Based Intelligent Information and Engineering Systems*, (2006), 888-895.

[11] Jin-Long Wang and Shu-Yin Chiang, Dynamic Handover Scheme for WiMAX. *International Knowledge-Based and Intelligent Information and Engineering Systems*, (2009), 729-735.

[12] Jian-Bo Chen and Tsang-Long Pao, Designing the Burst Web Load Balancing Architecture Using Fuzzy Decision, *International Journal of Innovative Computing, Information and Control*, Vol.5, No.6, (2009), 1689-1698.

[13] Q. Lili, V. N. Padmanabhan, and G.M. Voelker, On the Placement of Web Server Replicas, *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, (2001), 1587-1596.

[14] B. Li, M. J. Golin, G. F. Ialiano, and X. Deng, On the Optimal Placement of Web Proxies in the Internet, *INFOCOM*, (1999), 1282-1290.

Jian-Bo Chen received his MS from the department of Electrical Engineering at National Taiwan University, Taipei, Taiwan, Republic of China, in 1995, and his PhD from the Department of Computer Science and Engineering at Tatung University, Taipei, Taiwan, Republic of China, in 2008. He is currently an Assistant Professor in the Department of Information and Telecommunications Engineering at Ming Chuan University, Taoyuan, Taiwan, Republic of China. His research interests include network management, network security, and load balance.