WEB CLUSTER LOAD BALANCING VIA GENETIC-FUZZY BASED ALGORITHM

Chin Wen Cheong and Amy Lim Hui Lan

Faculty of Information Technology, Multimedia University
63100 Cyberjaya, Selangor Darul Ehsan, Malaysia.

Email: wcchin@mmu.edu.my

ABSTRACT

In this paper, genetic-fuzzy based Generalized Dimension Exchange(GDE) method is proposed to uniformly distribute the unprecedented Web cluster workload. Fuzzy set theory is used to capture the vagueness of the workload during redistribution period. According to the experts' subjective evaluations, a fuzzy inference system is established to aggregate the fuzzy web performance metrics into a so-called load-weight index which indicates the servers' workloads intensity. Based on the load-weight index, the genetic-fuzzy algorithm is applied to equally redistribute the workload among the servers. Finally, a simulation of 20 load-weight indices in a topology of 3-cube form Web cluster is implemented to illustrate the functionality of the proposed method.

Keywords: genetic algorithm, fuzzy set theory, generalized dimension exchange, fuzzy inference system, web cluster.

1.0 INTRODUCTION

Owing to the high demands from Web clients, Web server administrators sometimes replicate the information resources by establishing a Web cluster. A Web cluster consists of a collection of mirror servers that allow a Website to deliver information over the Internet (Menasce, 1998). Especially during the peak hours, the busty web traffic (Crovella and Bestavros, 1995; Leland and Taqqu, 1993) might cause drastic traffic congestion. The improper workload distribution sometimes triggers overloading in a web cluster. Consequently a dynamic load balancing method should be proposed in order to overcome misallocation of resources and low cluster servers utilization issues.

Round Robin DNS is one of the well-known methods that is applied in Web cluster load balancing and load sharing. However, for self-similar (Leland and Taqqu, 1993) arrival of requests, traditional Round Robin method is only practical for the uniformly distributed arrival requests (Colajanni et al., 1997). Additionally, the limited DNS scalability (Cardellini et al., 1999) of the reached workload has failed to perform Web cluster optimal resources utilization. To overcome the drawback of traditional DNS approach, studies have been carried out to improve the web server load balancing. Shaout et al. (1998) presented a batch job scheduler for a distributed system, which utilized a fuzzy logic algorithm to balance the load on individual processors and maximise the throughput to the overall system. Abani et al. (1993) formulated the load-balancing problem with fuzzy sets. A fuzzy decision model for the control and management of large decentralized system is more appropriate than the conventional probabilistic models. On the other hand, Park and Kuhl (1995) proposed a new approach to characterize the global state of uncertainty inherent in a large distribution system in terms of the fuzzy set theory and presents a fuzzy-based distributed load balancing algorithm that explicitly reflects the effect of the uncertainty in the decision making process.

Greene (2001) introduced a genetic algorithm (GA) scheduling routine, which with often, relatively low cost, finds well-balanced schedules. Yang and Wang (2004) suggested that routing with load balance and time delay based on genetic algorithm can solve problems of dynamic routing and can be applied in Internet especially in next hop dynamic route. As a result, the applications of artificial intelligence approach has become more popular due to its efficiency and accuracy compared to the traditional methods.

In this paper, the Generalized Dimension Exchange (GDE) method (Xu and Lau, 1997) is implemented in parallel computing to equalize the workload between multiprocessors. Each of the redundant servers in the Web cluster is considered analogous to a processor in multicomputer since processor is one of the essential components of a Web server. The connector between the servers is chosen based on the system desired bandwidth capacity requirement. In GDE method implementation, edge-colouring approach is utilized to pair the Web servers with respect to workload distribution. Arbitrary topology of the Web cluster is represented by a graph with its associated vertices. Minimum index orders of dimension are coloured using Brelaz Colour-Degree Algorithm (West, 1996) in order to accelerate the stable stage. The dimension of workload exchange among the servers is based on the coloured graph and fuzzy system with GA approach.

A comprehensive workload understanding is essential in order to study the web traffic. Abini (1993) quantified the load intensity as the number of requests in waiting queue where a high load demands transfer of a job, and a low load node can accept a request. However, the simplification of the load indication is insufficient to capture the overall load characteristic which includes the arrival rate of the request, their total response time, hardware capability and etc (Cai et al., 1997). According to Zaki et al. (1996), load-balancing system is modelled by considering the hardware performance, data size, number of work per iteration, total cost of redistribution and etc. Nevertheless, the huge monitoring of parameters has caused unavoidable communication delay and costs. It is caused by the unstable network condition, random surfing user behaviour and inefficiency of the workload handling approach. Due to the nature of the workload uncertainty during workload exchange, fuzzy approach has been selected to model the workload.

In this study, three basic metrics measurement such requests average size(ωsize), number of request in queue(ωnumber) as well as communication overhead(ωcom) are used to determine the so-called load-weight(ω). Each of the metrics parameter is represented by fuzzy linguistic term with predefined membership functions. Fuzzy rules are established under experts' subjective judgements (Ramachandran and Sankaranarayanan (1991) where the fuzzy inference system (FIS) is aggregated with the outputs fuzzy sets of each rule into a single fuzzy set. In addition, genetic algorithm robust searching capability is used to optimally select the workload for each of the servers during workload exchange. During the searching process, binary string represents the existence of certain workload in the server and the fittest genome according to the predefined fitness function is survived as the selection to determine the workload allocation.

2.0 GENETIC-FUZZY BASED GDE APPROACH IN LOAD BALANCING

The aim of this method is to compare and equalize each neighbour server's workload in an arbitrary Web cluster topology until the workload distribution reaches a balanced stage. The first step of GDE approach is to determine the dimension indices by using the edge-colouring. The dimension (Xu and Lau, 1997) is defined as edges with the same colour while the iterative process for all dimensions in the corresponding system topology is defined as a sweep. Based on the predefined dimensions, neighbour server workload exchange is equalized along the order dimension. FIS is used to determine the load-weight by aggregating three predefined parameters(ωsize, ωnumber, ωcom).

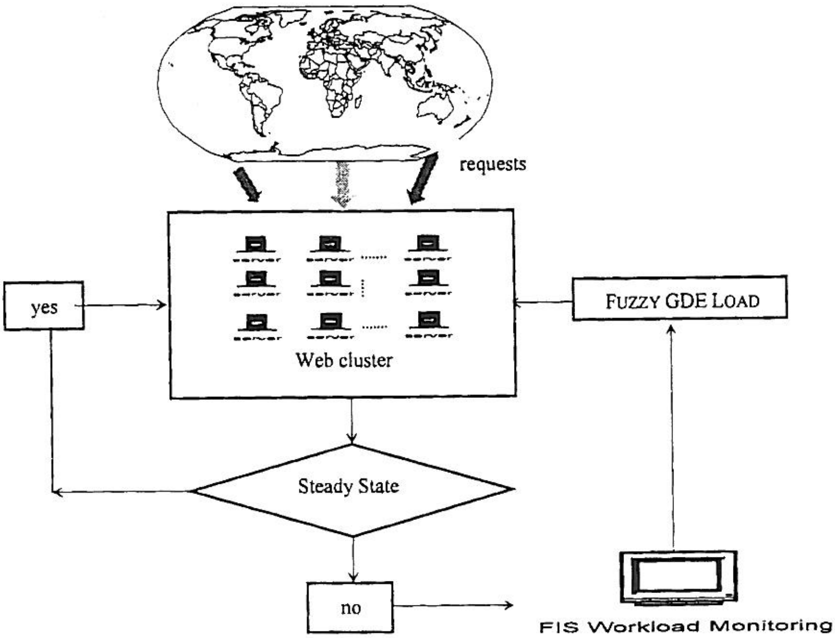


Fig. 1 : Process flow

Optimal workload distribution among the pair servers is determined using FIS and genetic algorithm approach to accelerate the system to achieve the stable stage. The structure of the process flow is illustrated in Fig. 1.

2.1 Web Cluster System Modelling

A given arbitrary graph *G(V,E)*, represents a Web cluster where V and E are denoted as the vertex and the edge of a graph respectively. For an arbitrary link oriented network, a vertex *(V)* represents a server and the edge denotes the connection between adjacent for an identical servers pair in the model.

If the edge colouring of graph *G(Vi,Ej)* is assigned with k colours, the edges of an identical graph *G(Vi,Ej)* are coloured by a minimum number of colours and the adjoining edges are in different colours. The chromatic index of the graph *G(Vi,Ej)* is denoted by $\psi(G)$ which is based on Vizing's theorem (West,1996), the equivalent $\psi(G)=\Delta$ is defined as class I and $\psi(G)=\Delta+1$ is known as class II where Δ is the maximum degree of vertex of a graph. The chromatic index, $\psi(G)$ for the edge between x and y is defined by a 3-tuple (x, y, ψ). In this paper, the edge-colouring is determined as vertex colouring associated with each graph *G(Vi,Ej)* to its line graph L(G). Hence, the chromatic index of edge colouring is equal to the chromatic number of vertices where $\psi(G(Vi,Ej)) = \Psi(L(G(Vi,Ej)))$.

The first stage of the computation is to identify the Web cluster workload distribution. FIS with predefined rules is implied an index known as load-weight(ω). System distance (Bharat et al., 1996) is a measurement metric that evaluates the efficiency of the current Web server workload. The workload-exchange termination is based on the following criteria:

a. when the total servers' load-weight is less than a predefined parameter,ξ and the system balance state is confirmed based on the following equation
   $\Sigma |\ \omega x - \omega mean\ | < \xi$
   where ξ is the predefined system's distance threshold with respect to the system's overall distance which are governed by the topology and the size of the system
b. when all the servers are idle.

2.2 Web Cluster System Colouring

The occurrences of unequal workload distribution among Web servers lead to the *GDE* dynamic load balancing. Brelaz Colour-Degree Algorithm (West, 1996) approach is used in order to colour an arbitrary given graph *G(Vi,Ej)*. The vertex colouring algorithm divide the edge based on the least colour degree and provide sequentially the precise chromatic number of a graph, $\Psi(L(G(Vi,Ej)))$. The algorithm is expressed as follows:

Step 1: For a given arbitrary graph *G(Vi,Ej),* order the vertices in the form of decreasing order of degrees.
Step 2: The maximum degree vertex is coloured with colour Ca.
Step 3: Choose the largest colour-degree vertex and if the vertex isconnected to some other vertex/vertices, select any vertex of maximum degree in the uncoloured subgraph.
Step 4: With the minimum possible colour, colour the chosen vertex from Step 3.
Step 5: After all the vertices have been coloured, end the colouring algorithm Or else go to Step 3.

Based on the vertex colouring, the edge colours can be determined accordingly before the workload is exchanged among the neighbour servers.

3.0 WEB CLUSTER FUZZY WORKLOAD MODELLING

The additional servers in a Web cluster are indicated with the load-weight index which represent the workload intensity. In general, for certain time unit, T, the load-weight of each server is determined by alternating Web performance metrics that characterize the workload intensity. Workload intensity of a particular server is specified by 3-tuples such as

$$\omega = \{\omega size, \omega number, \omega com\}$$

where

i. ω represents the load-weight in a server;

ii. ω_{size} represents the average file size in a certain server;

iii. ω_{number} represents the number of waiting load in a certain server.

ω number is the combination of finite set of load $f_1, f_2, f_3, \ldots, f_M$ in the Web cluster being considered where M is the total number of load in the Web cluster. A Web server's (K_x) ω_{number} is illustrated as

$$\omega_{number}\ (K_x) = \{f_1, f_2, f_3, \ldots\ldots, f_N\}$$

where N is the total number of load in a particular server. ωnumber is the variation for every server in certain time-unit which is represented by independent subsets. Let S1, S2, S3, S4, ..., SP be the disjointed subsets of ωnumber in a particular server(K1, K2, K3, ..., Kp) such that no two subsets Sm and Sn (m≠n) are identical i.e as well as with the condition fm ≠ fn.

$S_m \cap S_n = \phi$ and $S_1 \cup S_2 \cup \ldots \cup S_m \cup S_n \ldots \cup S_M = \omega_{total\_number}$

where M is the total number of subsets, ω_{total_number} is the total number of the loads in time T.

iv. ω_{com} represents the overall communication cost and delay between 2 servers.

For the sake of simplicity, the communication cost is stated as:

$$\omega_{com} \propto N_{workload}$$

where $N_{workload}$ is the number of workload has to be immigrated among the paired-server in the Web cluster.

Due to the vagueness of the distributed system, fuzzy approach is implemented to determine the load-weight(ω). Fuzzy inference system (Jang et al., 1997) is based on fuzzy set theory and fuzzy rules-based approaches in decision analysis and variety of fields, especially dealing with uncertain and complex systems. The portability of fuzzy system allows human linguistic language approach to determine the grade of the membership as well as the fuzzy rules for the inference engine. Fuzzy inference system (FIS) involves some procedures to consider whether it is a fuzzy or crisp result depending on the user's intention. The components of a fuzzy system includes fuzzification (fuzzy input memberships), inference engine and defuzzification respectively.

According to experts' subjective evaluation, each of the tuples is illustrated with fuzzy linguistic term and its associated membership function. For instance, the workload metrics' values are fuzzified into three fuzzy linguistic spaces as following:

$$\omega_{size} = \{Small, Medium, Large\}$$
$$\omega_{number} = \{Short, Fair, Long\}$$
$$\omega_{com} = \{Low, Moderate, High\}$$

The rule which decides the load-weight values are shown in Table 1.

Table 1: Fuzzy rules

IF					THEN
ω_{size}	OPERATOR	ω_{number}	OPERATOR	ω_{com}	ω (LOAD –WEIGHT)
LARGE	AND	LONG	AND	HIGH	EXTREMELY HEAVY
LARGE	AND	LONG	AND	MODERATE	VERY HEAVY
LARGE	AND	LONG	AND	LOW	HEAVY
LARGE	AND	FAIR	AND	HIGH	VERY HEAVY
LARGE	AND	FAIR	AND	MODERATE	HEAVY
SMALL	AND	FAIR	AND	LOW	VERY LIGHT
SMALL	AND	SHORT	AND	HIGH	LIGHT
SMALL	AND	SHORT	AND	MODERATE	VERY LIGHT
SMALL	AND	SHORT	AND	LOW	EXTREMELY LIGHT

Fig. 2, Illustrates the linguistic input and output of the model system

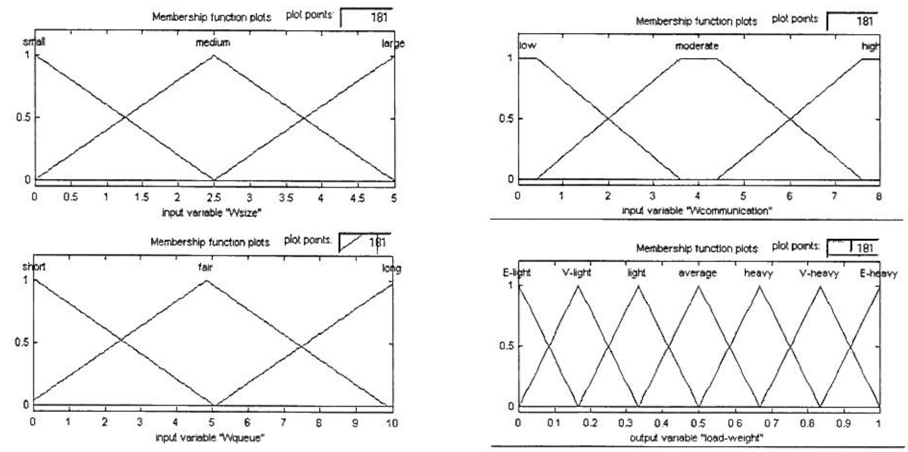


Fig. 2. Parameters and associate memberships function

FIS is used as a max-min composition in the calculation of load-weight value. Fuzzy inference with predefined rules integrate all the 3-tuples intensity that finalizes a load-weight with the following seven linguistic term value: Extremely Heavy, Very Heavy, Heavy, Fair, Low, Very Low, Extremely Low.

4.0 GENETIC ALGORITHM APPROACH IN WORKLOAD OPTIMAL ALLOCATION

Genetic algorithm (GA) is introduced by Goldberg (1989) as an optimization search algorithm. The first step of GA is to generate an initial population and formulate a string pattern for the complete solution of the particular model. The chromosome is represented by a binary string that depends upon the condition of the system. Every bit represented the existence of a load (f_p) in a particular server($\mathbf{K}_a$) with the condition each load only allows to exist once among the servers. By applying the operators such as selection, crossover and mutation, the chromosome with the highest fitness is chosen to determine the next time period workload allocation. The major parameters involved in GA are

i. Population size of the chromosomes
ii. Maximum number of generation
iii. Probability of crossover (P_{cross}) and
iv. Probability of mutation (P_{mute}).

Step 1: Initialization
i. set population size and maximum generation
ii. set P_{cross} and P_{mute}
iii. set generation = 0
iv. set the bit length for each parameter

Step 2: Generation of initial population
Based on the Web server (K_a) load quantity, initial the binary bit where 1 represents the existence and 0 non-existence of the load in the server.

Step 3: Evaluate the fitness function for each of the string in the current population according to the fitness function. The fitness function is governed by 3 parameters namely the average file size, number of queuing and communication cost. By using the predefined FIS, the fitness function is determine by the following function:

$$Fitness\ Function = \frac{1}{|\omega_x - \omega_y|}$$

If the Fitness Function $>\mu$, terminate the searching process, where μ is the predefined fitness function threshold. Else go to *Step* 4.

Step 4: Set offspring count = 0 and generate the operator according to the crossover rate, mutation rate. Perform crossover with probability Pcross. If crossover is not performed, put chromosome into the next generation and go to Step 3. Otherwise
a. Select mate from population with uniform probability.
b. Select crossover point between the string with uniform probability.

Step 5: Repeat the Step 3 if the fitness function threshold μ is not fulfilled.

Step 6: If for consecutive n-th generation the μ is not fulfilled, choose the fittest chromosome as the final solution.

4.1 Numerical Example

Consider two servers with the properties as illustrated in Fig. 3, the details of the GA approach and fuzzy system is shown in Table 2.

Table 2: Workload allocation in pair-servers,K1 and K2

		1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	Fitness
1st	K_1	1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0	2.18
	K_2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
2nd	K_1	1 0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0	11.63
	K_2	0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	
3rd	K_1	1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0	4.08
	K_2	0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0	
n-th	K_1	1 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0	30.30
	K_2	0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0	

Assume after n-th searching processes, the selected distribution of the pair-servers is achieved with the fitness 30.

5.0 FUZZY BASED GDE IMPLEMENTATION

After the determination of the $G(\psi)$, consider two neighbour servers(x, y) with ω_x and ω_y representing the initial load-weight resulting from the FIS in server x and y respectively. Each load(fp) is indicated with a reference integer and respective file size(f_{size}) with the condition $f_m \neq f_n$, $S_m \neq S_n$ and f_p can only exist in a server. Subsequently, workload distribution involves a number of iteration sweeps to reach the uniform distribution for each server. System distance as a measurement of the system stability is normalized (Bharat,1996) with the respect maximum possible system distance. A normalized system distance threshold is defined as ξ. The workload interchange between servers is resolved by using the subsequent algorithm:

Step 1: Identify every servers'($\mathbf{K}_i$) workload intensity by using the FIS for time period t.

Step 2: If the server detects non-existence of steady state, start iterative sweeps in n dimension for the same color edges with the existence of the edge (x,y) and with color n where $n=1$, $n \leq \psi$.

Step 3: Using GA algorithm(3.1) randomly reallocate the workload in the involved server-pair and FIS will determine the workload allocation by concerning the three predefined metrics.

Step 4: After one dimension, detect the system state based on the following equation for every server K.

$$\sum_{i=x,y}^{k} \left| \omega_i^{t+1} - \omega_{mean}^{t+1} \right| < \xi$$

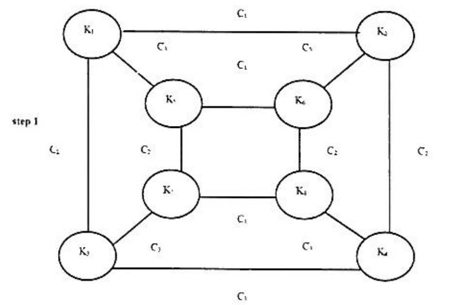
If the steady state has not been achieved, go to Step 2. For the next dimension, n=n+1, repeat Step 2 and Step 3 ($n \leq \psi$) until a full sweep has been proceeded. Else, terminate the distribution algorithm.

5.1 Numerical Illustration

A Web cluster is chosen arbitrarily, where the topology of the internal servers is in the 3-cube form and the Web cluster consists of 8 servers. The edge graph is associated with its line graph and hence the edge colouring can be

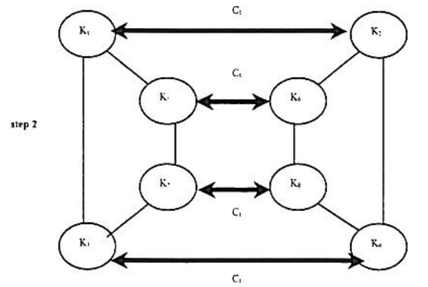
Table 3: Workload initialization

$i(f_i)$	Size(MB)	$i(f_i)$	Size(MB)
1	3.56	11	2.78
2	3.21	12	4.88
3	2.51	13	0.65
4	4.88	14	4.22
5	3.50	15	3.21
6	3.56	16	4.55
7	0.80	17	2.20
8	1.20	18	4.33
9	3.22	19	1.11
10	0.98	20	4.30



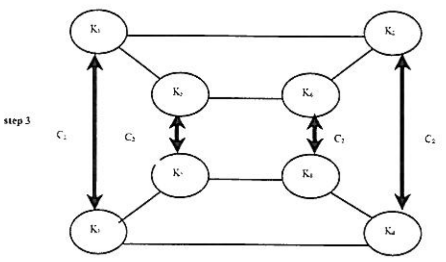
Server	File allocation $i(f_i)$	W (Load-weight)	W – Wmean (Wmean=0.1339)
K_1	1, 2, 3, 4, 5, 6, 7, 8, 9,10	0.521-very heavy	0.339625
K_2	0	0.063 -extremely light	0.118375
K_3	0	0.063 -extremely light	0.118375
K_4	0	0.063 -extremely light	0.118375
K_5	0	0.063 -extremely light	0.118375
K_6	0	0.063 -extremely light	0.118375
K_7	0	0.063 -extremely light	0.118375
K_8	11, 12, 13, 14, 15, 16, 17, 18, 19, 20	0.552 -very heavy	0.370625

Initially State: **Normalized Distance = 1.00**



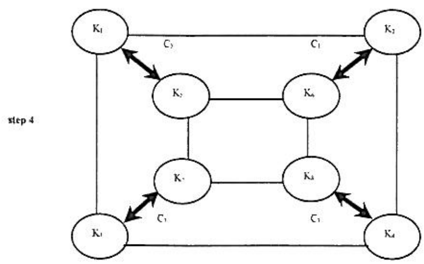
Server	File allocation $i(f_i)$	W (Load-weight)	W – Wmean (Wmean=0.1728)
K_1	1,4,5,6,7,9	0.437-heavy	0.22125
K_2	2,3,8,10	0.285-light	0.06930
K_3	0	0.063-extremely light	0.15275
K_4	0	0.063-extremely light	0.15275
K_5	0	0.063-extremely light	0.15275
K_6	0	0.063-extremely light	0.15275
K_7	14,15,16,18, 19,20	0.448-heavy	0.23225
K_8	11,12,13,17	0.304-average	0.08825

First Dimension: **Normalized Distance = 0.86**



Server	File allocation $i(f_i)$	W (Load-weight)	W – Wmean (Wmean=0.2695)
K_1	1,5,6,9	0.358-heavy	0.05675
K_2	2,3	0.273-light	0.02825
K_3	4,7	0.171-light	0.03025
K_4	8,10	0.219-very light	0.08225
K_5	15,19,20	0.288-light	0.01325
K_6	11,13	0.299-average	0.00225
K_7	14,16,18	0.385-heavy	0.08375
K_8	12,17	0.317-average	0.01575

Second Dimension: **Normalized Distance = 0.22**



Server	File allocation $i(f_i)$	W (Load-weight)	W – Wmean (Wmean=0.2794)
K_1	5,6,9,19	0.322-average	0.0120
K_2	2,3	0.273-light	0.0370
K_3	4,7,14	0.323-light	0.0130
K_4	8,10	0.219-very light	0.0910
K_5	1,15,20	0.336-average	0.0260
K_6	11,13	0.299-average	0.0110
K_7	16,18	0.391-heavy	0.0810
K_8	12,17	0.317-average	0.0070

Third Dimension: **Normalized Distance = 0.20**

Fig. 3: Load balancing on a 3-cube Web cluster system

considered as a vertex colouring. After the determination of Brelaz Colour-Degree Algorithm, the hypercube structure is coloured with 3 different colours, c1, c2 and c3. The initial workload of the eight servers is arbitrarily defined respectively with the associated vertices K1, K2, K3, K4, K5, K6, K7 and K8 as illustrated in Table 3.

Initially for genetic algorithm approach, the genome length of the encoded chromosomes is represented by 20 randomly distributed binary values where every load is represented by 1 binary bit. The population size, P_{cross}, P_{mute}, μ, number of crossing site and the numbers selected fittest chromosomes are fixed to 0.5, 0.1, 0.30, 1 and 1, respectively. The normalized system distance threshold of the workload is fixed as $\leq \xi$.(where ξ. is equivalent to 0.20).

The initial workload distribution process is terminated once the fitness chromosome reached the stationary state for 100 consecutive generations. After the termination, the previous distributed workload will proceed new workload redistribution. The above-mentioned procedure is repeated until the fixed threshold function is fulfilled. The details of the workload exchanged among the servers are illustrated in Fig. 3.

6.0 CONCLUSION

In this paper, fuzzy approaches in workload modelling and workload intensity determination showed the enhancement of the representativeness of the model and ability to capture the vagueness of a Web cluster workload distribution system. These approaches provide non-trivial information and techniques especially to network administrators who intend to manage their network capacity. However, the overall workload characteristics can be further improved with the information such as the arrival rate of workload in each server, execution time, hardware as well as software requirements. In future studies, we would like to include these criteria in the server cluster load balancing analysis.

REFERENCES

Abini, K., Akingbehin, K. Shaout, A. (1993). Fuzzy Decision Making for Load Balancing in a Distributed System. Proceedings of the 36th Midwest Symposium on Circuits and Systems (pp. 500 -502).

Bharat, S. J., Hosseini, S. and Vairavan, K. (1996). Stability Analysis of a Load Balancing Algorithm. Proc. of the IEEE Southeast Symposium on System Theory (pp. 412-415) Washington, DC, USA

Cai, W.T., Sung, B., Heng, A. & Li, Z.(1997). A Simulation Study of Dynamic Load Balancing for Network-based Parallel Processing. Proceeding of Parallel Architectures, Algorithms, and Networks(pp. 383-389). Washington, DC, USA.

Cardellini,V., Colajanni, M. and Yu, P.S. (1999). DNS Dispatching Algorithms with State Estimators for Scalable Web Server Clusters, World Wide Web J., 2(3), 101-113.

Crovella, M., and Bestavros, A. (1995). Explaining World Wide Web Traffic Self-Similarity. Tech. Rep.BUCS-TR-95F-015, Voston University, CD Dept, Boston MA 02215.

Colajanni, M., Yu, P.S., & Dias, D.M.(1997). Scheduling Algorithms for Distributed Web Servers, Proceeding of. ICDCS'97 (pp. 169-176). Baltimore, MD.