ISSN:0975-9646

Squid Proxy Server Cache Management using K-means Algorithm

Subhash Chand[1] and Sanjay Mathur[2]

[1] *Department of Computer Engg.*
[2.] *Department of Electronics & Communication Engg.*
College of Technology, G B Pant University of Agriculture & Technology, Pantnagar-263 145 (India)

***Abstract*— : All services needs to have an accuracy, availability and speed. Internet users are increasing very rapidly which indicates a challenge in maintaining the accuracy, availability and speed in the Internet services. To reduce the load on Internet and improve the Quality of Service (QoS) the web caching technique are in use and all proxy server are maintaining their own cache to provide the requested web page locally to the user. In this paper K-means algorithm is used to identify the web object from the cache of proxy server for its removal and making the room for new web object. It is observed that K-means algorithm always gives better results to improve the HIT rate(i.e. requested web object is available in the cache of proxy server) and remove the unwanted web objects from the memory of proxy server to make the room for new cached web objects.**

***Keywords*—** Put your keywords here, keywords are separated by comma.

I. Introduction

The network of networks becomes the road or means of transport for exchange the data, information, etc. for development, growth of human being especially in the fields of social life, research, etc. Internet is the fastest way of communication and definitely it consumes the resources like memory, CPU time, bandwidth etc. The resources are limited in capacity and numbers. Therefore optimal use of the resources becomes a challenge on which researchers are working. The proxy server have limited memory to store the cached web objects and limited size capacity of internet bandwidth to access the web pages from remote server. Proxy server retains some web pages in its memory and return a HIT, if this client requested web page available in the cache memory of proxy server, otherwise MISS is returned. The web pages available in cache but not requested by the clients for a long time are removed from the cache memory of the server to free the memory for new web objects. If proxy server returns the maximum HIT then internet bandwidth will be saved accordingly and load on the original web server will also be reduced.

Therefore our main objective in cache management should be to increase the percentage of HIT, reduce the percentage of MISS and reduce the average memory size used by the cached web objects by removing the unused web objects form the memory of the proxy server. To work on this we use one day data of one proxy server stored in access.log file of the proxy server. This file contains all the information about the user requested web pages. The users are working as a node of LAN and accessing the Internet through a proxy server and after fulfill the user request proxy server stores all the information about the requested web page in the file access.log. This file is written by operating system in append mode. This proxy server was configured on centos 5.4 operating system having squid proxy server version 2.

The problems of overloading of web servers and the congestion on the network becomes very serious and need special consideration for its solution. Web caching was introduced in 1990s to help decrease network traffic and reduce load on original web server by storing the copies of web objects onto end user machine or proxy server memory [1]. Using web caching techniques the problem of congestion on network can be solved up to a certain level and many researchers are working to improve the web caching technique and get the better results. Modern proxy servers have a significant document miss rate typically from 40 to 70 percent [2]. Proxy servers are useful in providing the security to the clients, reduce load on web server, decrease traffic on network, reduce client retrieval time, etc. Proxy web caching architectures can be divided into three major categories i.e. hierarchical, distributive and hybrid [3]. Many cache replacement strategies have been developed and may be divided into three category: Traditional Policies (Least Frequently Used (LFU), Least Recently Used (LRU)), Key-based Policies (LRU-MIN, LRU-Threshold, Lowest-Latency First), Cost-based Policies (Greed Dual Size, Hierarchical Greedy Dual) [4]. Web caching has some advantages. It reduces bandwidth consumption, decreases network traffic and congestion, reduces access latency, reduces workload on original web server. Web caching also has some disadvantages, e.g. end users might get stale data, checking the requested web page into cache is an overhead, consuming memory of server, reducing the hit on original web server [4]. A study on the overheads associated with disk I/O for web proxies is made and proposed Web-Conscious Storage Management technique for high performance under web workloads [6]. The differentiated content caching services as a key element of internet content distribution architecture implemented and resource management used to achieve performance differentiation in proxy caches [7].

There has been number of approaches reported in literature which have used neural networks and fuzzy systems for cache management. In [8] an approach splitting the client side cache into two caches, short term cache and long term cache was reported. This approach improved the Byte Hit Ratio (BHR) and Latency Saving Ratio (LSR). In [9] an approach compared with Least Recently Used (LRU), Least Frequently Used (LFU) and optimal case. Neural networks achieve very high hit rates in particular case. In [10] a neural network based cache replacement policy which effectively identify and eliminate inactive cache have a batter performance as compared to the conventional schemes.

It is seen that neural networks can be effectively applied to develop cache management systems. In our approach, for making the decision whether to include the web object in the cache or not is supported by k-means algorithm, which is a well known unsupervised neural technique.

The proxy server stores all the information about the user requested web pages in the file access.log which has the default location in centos operating system as /var/log/squid/access.log. This file is written by operating system in an append mode. Squid does not impose size limit on access.log file but some operating system have a maximum file size limit [5]. In total, ten (10) attributes about the requested web pages are stored in this file. These attributes are Date and Time stamp, time taken to retrieve the web object, client IP address; web page hit status, web page data size, server connection method, name of requested web page, user authentication, IP address of web server and requested document type. Out of these 10 attributes 4 attributes, name of requested web page i.e. URL, retrieval time, data size and web page hit / miss status are selected from log file for analysis. One more attribute know as frequency is calculated by counting the repetition of each web page in the log file. Finally five (5) attributes are extracted from the one-day log having 398630 records. These five attributes are converted to numerical values before applying the clustering method to cluster the data into different number of clusters or groups. The data of these clusters analyzed and take a decision on these clustered data to improve the hit rate reduce the miss rate and average memory size used by the cached web object. Smaller size clusters / groups are considered as outliers web objects and removed from the memory of proxy server. After removing these outliers from memory the percentage of hit rate, percentage of miss rate and average data size is calculated. This process is repeated ten times and the graph between number of clusters and percentage of hit rate, percentage of miss rate and average data size are plotted to see the final result. The graph shows that percentage of hit rate is increased with the number of clusters and decreases the percentage of miss rate and average data size with the number of clusters. The TCP HIT and MISS, are further categorized as TCP_CLIENT_REFRESH_MISS,TCP_IMS_HIT,TCP_MEM_HIT, TCP_NEGATIVE_HIT, TCP_REFRESH_HIT, TCP_REFRESH_MISS, etc.[5]. These subcategories are considered in the respective main category as HIT, MISS.

This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

II. Web page Caching

Caching is the method to retain the web pages on the local memory of computer / server. The identification of web objects to retain it into cache or not is based on some characteristics of web objects like recency (time information about related object last requested), frequency (number of times a web object visited), size (size of web object in bytes), online transactions (should not cache), web queries (should not cache), etc. If cached web page is visited in future then a copy of the stored web page is made available to the user from the cache not from the internet. Therefore internet bandwidth is saved because the web page is retrieved from local memory. The cache is maintained separately on the client computer and on the server known as proxy server. The user requested web page is first searched on the user computer cache; if it is not found in the cache then it is searched on the proxy server cache on LAN. If it is not found at both the locations then desired web page request is forwarded on Internet by proxy server to retrieve the desired web page from the original web server if available, otherwise an error message is generated to the user by the proxy server. Therefore retaining the web pages in the cache definitely saves the internet bandwidth and reduces the retrieval time and the load on the original web server. Some advantages of web caching are :

1. Provides fast access to the cached object
2. Reduces the traffic on the network therefore save the internet bandwidth
3. Reduces the load on original web server
4. Improves the failure tolerance and robustness of whole web system.

Apart from these advantages web caching also has some disadvantages. Some disadvantages of web caching are:

1. If web object not found in the cache then its retrieval time will be more by an additional of time to search it in the cache
2. It is possible to get a stale copy of web object
3. It consume the memory of client or proxy server or both
4. CPU time consumed in searching of requested web page into cache.

It is very important to decide what to be cached or not. We have the limited size of memory to retain the cached object. The number of web pages available on the Internet which may be cached is very large. Therefore it is not possible to cache all the web pages. This indicates that frequently accessed web pages must be kept in the cache of proxy server so that percentage of hit ratio must improve. If we have free memory then there is no problem to retain the new cached web object into memory, otherwise one or more web objects stored into memory must be sacrificed to create the room for new cached web object. Identification of web objects deleted from the memory of proxy server is very important and needs a decision system to decide whether to retain the web page in the cache or not.

III. K-MEANS ALGORITHM

K-means algorithm is employed for clustering. Clustering is unsupervised classification. It is also known as grouping of data points that are closed to each other. Clustering is based on the distance matrix between data points. For a given value of k, partition n observations into k clusters that optimize the partition/clustering criteria. It is unsupervised clustering algorithm in which k indicates the number of clusters. Normally k is defined by the user and n is the total data set.

K-means clustering partitions the n observations into k groups / cluster, where k ≤ n. These k clusters are denoted by c1, c2, c3, ….. ck and represented collectively as C = { c1, c2, c3, ….. ck }. For a given set of n input vectors (x1, x2, x3, ….. xn), where each input vectors is of dimension d, K-means clustering minimize the sum of squares with in the cluster

$$\underset{C}{minimize} \sum_{i=1}^{k} \sum_{x_j \in c_i} \left\| x_j - \mu_i \right\|^2 \quad \text{........(1)}$$

where μi is the mean of points in Ci.

In our case n was 7392 and k was chosen 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 in different steps.

IV. SQUID PROXY SERVER CONFIGURATION AND HARDWARE USED

The server installation and configuration play an important role in storing and retrieving the logs. Following installation and configuration of the actual proxy server is used. We use the Dell Power Edge 2900 machine as server having the following configuration:

Processor : 1.86 GHz Quad-Core (one number)
Bus speed : 1066 MHz
Cache : 2x4 MB L2
System Memory : 2.0 GB
System Memory Speed: 667 MHz
Secondary Memory: 146 GB, 15K rpm (two number)

Cent OS 5.4 (32 bit) is used as operating system having Red Hat Nash Version 5.1.19.6. The file system was ext2 and at the time of installation the following mount points have been created

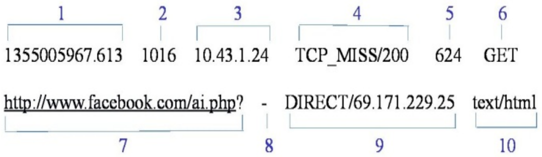
/	Maximum allowable size (all remaining space allotted)
/boot	2 GB
/var	80 GB
/temp	20 GB
Swap	4 GB

Squid proxy server is installed at the time of installation of operating system. The default location of main squid configuration file is /etc/squid/squid.conf. The default configuration file is installed at the time of installation which is changed as per user's and network's need.

The squid –z command used at command prompt to create the cache. This command will create the 16416 directories under the path /var/spool/squid. User authentication and blocking of websites is not used on proxy web server. Therefore users can access all the websites on Internet and having the same Internet access rights.

V. DATA AND ITS PREPARATION FOR ANALYSIS

The squid proxy server generally stores the log files in the directory /var/log/squid. Three types of log files namely access.log, cache.log and store.log are stored in this directory. All the web requests completed by proxy server are stored in access.log file. The data are appended at the bottom of file and the size of this file increases always. After a certain size of this file a backup file access.tar.gz is created automatically and new access.log file comes into existence. Given below is the example of one record of this file. Each record contains ten fields namely 1 to 10.



```
1355005967.613  1016  10.43.1.24  TCP_MISS/200  624  GET
http://www.facebook.com/ai.php?  -  DIRECT/69.171.229.25  text/html
```

Description of ten fields is given in the table 1 :

Table1: Description of ten fields of log file record

Field No.	Meaning	Description
1	Time stamp	Date and Time Stamp when a URI mentioned at no. 7 is requested by a user.
2	Duration	Time consumed between sending the request to proxy and completing the request by the proxy in milliseconds.
3	Client address	Client machine IP Address, which made the request to proxy server.
4	Result codes	There are two fields separated by /. First is the cache result and second is the HTTP result code.
5	Bytes	Total amount of data delivered to the client.
6	Request method	Method used to get the requested object. i.e. GET, PUT, etc.
7	URL	Uniform Resource Locator requested by a user.
8	RFC931	Detail of a user authenticated by server before request is made. In case of no authentication – is recorded in log file.
9	Hierarchy code	Normally it contains two fields separated by /. First indicate how the request was handled and second indicates the IP address where the request was forwarded in case of MISS.
10	Type	Type of the content requested by the client. i.e. text, image, etc.

Sample data taken from access.log file and used in analysis is given as follows:

```
1355005967.613 575 10.42.1.134 TCP_MISS/200 3279 GET
 http://googleads.g.doubleclick.net/pagead/ads? –
DIRECT/74.125.236.13 text/javascript

1355005967.614 552 10.42.1.134 TCP_MISS/200 7017 GET
http://sim.in.com/2/1b292b9f1513da8b6293d1a6cebfdabb_ls_lt
.jpg - DIRECT/72.246.188.163 image/jpeg

1355005967.630 568 10.42.1.134 TCP_REFRESH_HIT/200
 16211 GET http://pagead2.googlesyndication.com/pagead/
osd.js - DIRECT/74.125.236.13 text/javascript

1355005968.298 207 10.42.1.134 TCP_HIT/200 20657 GET
http://pagead2.googlesyndication.com/pagead/gadgets/
components/ryo.swf  - NONE/- application/x-shockwave-flash

1355005972.213 562 10.42.1.134 TCP_REFRESH_HIT/304
505 GET http://platform.twitter.com/widgets.js -
DIRECT/23.11.127.144 application/javascript

1355005972.750 404 10.42.1.134 TCP_REFRESH_HIT/200
 58656 GET http://connect.facebook.net/en_US/all.js -
 DIRECT/118.214.223.139 application/x-javascript
```

Field no. 1 is used to take one day data from the log file which contains 398630 records. Out of these 10 fields only 04 fields i.e. retrieval time in ms (field no. 2), Cache TCP_HIT or MISS status (field no. 4), requested URL / URI (field no. 7) , and web page size in bytes (field no. 5) are selected for analysis point of view. From URL field the prefix text known as http:// , https:// , www , etc. and postfix text i.e. text after first back slash '/' have been removed from each record of access.log file to concentrate on the main web pages only. For example consider the following URL's

http://googleads.g.doubleclick.net/pagead/ads?
http://sim.in.com/2/1b292b9f1513da8b6293d1a6cebfdabb_ls_lt.jpg
http://pagead2.googlesyndication.com/pagead/osd.js
http://pagead2.googlesyndication.com/pagead/gadgets/components/ryo.swf
http://platform.twitter.com/widgets.js
http://connect.facebook.net/en_US/all.js

After removal of pre and post text the URL's will look like

googleads.g.doubleclick.net
sim.in.com
pagead2.googlesyndication.com
pagead2.googlesyndication.com
platform.twitter.com
connect.facebook.net

It is observed that many URLs become duplicated. To replace the multiple occurrence of each URL by a single occurrence these duplicate URLs are identified and its number of occurrence or frequency is counted. Each URL is replaced by a code number and its number of occurrence by adding one more field known as frequency to the above said 04 fields. Finally we get the five attributes (URL code, frequency, duration, Hit / Miss, web page size) which are used in analysis. After adding the frequency attribute 398630 records were reduced to 7392 records. The hit and miss like TCP_CLIENT_REFRESH_MISS,TCP_IMS_HIT,TCP_MEM_HIT, TCP_NEGATIVE_HIT, TCP_REFRESH_HIT, TCP_REFRESH_MISS, etc are converted to the category as HIT, MISS and replaced by a code number. Finally all the data prepared in this fashion are converted into numeric form which are used for clustering. The final data sheet will look like as follows:

URL Code	Frequency	Duration (ms)	Hit / Miss Code	Web page size (Bytes)
1	1	1	1	2644
2	1	6692	1	39
3	1	401258	1	57301
4	862	5956	1	621
5	1	5362	3	1886

VI. IDENTIFICATION OF WEB PAGES TO REMOVE FROM CACHE

Initially it is assumed that all the URLs identified for analysis are kept into server cache memory. The task is to identify the web pages for removal from the cache so that memory of server will be free for new web pages. These URLs are known as outliers. It is also very important that after removal of these outliers the hit rate must increase and miss rate falls down. The average memory size used to cache objects must also go down.

To achieve higher hit rate and decrease the average memory size data are clustered into different number of groups to identify the outliers URL's. This process is repeated ten times. The details are given as follows:

Table 2: Number of clusters versus number of outliers

S.No.	Number of Clusters/ Groups created (A)	No. of URL's in small sized Cluster / groups (known as outliers) (B)	No. of URL's in big sized cluster / groups (known as cacheable objects) (C)
1.	10	216	398414
2.	20	293	398337
3.	30	1427	397203
4.	40	1609	397021
5.	50	1729	396901
6.	60	1751	396879
7.	70	1884	396746
8.	80	2282	396348
9.	90	2287	396343
10.	100	2296	396334

These outliers removed from the cache of proxy server and an observation on hit rate, miss rate and average memory size is made. Three graphs between number of cluster, hit rate, miss rate, and average memory size are drawn which indicate improvement in the percentage of hit rate and reduced cache memory size.

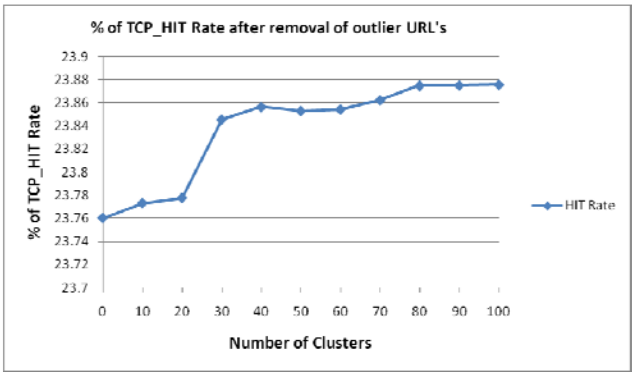


Fig.1. Graph between number of clusters and % of TCP_HIT rate.

The origin point indicates the initial state or prior clustering position of the data. After that data are clustered into 10, 20, …100 clusters and each time percentage of HIT rate is calculated. The graph in fig 1 indicates that percentage of HIT rate is increasing as the number of clustering increasing. The number of outlier URLs is also increasing with the number of clusters.

The graph as shown in fig 2 between number of clusters and percentage of MISS rate also indicates that, if percentage of HIT rate increases than percentage of MISS rate decreases. The fall down in the miss rate also indicates that more requested webpage is available in the cache of proxy server. Therefore percentage of hit rate is increased and traffic on the Internet, use of Internet bandwidth is also reduced proportionally.

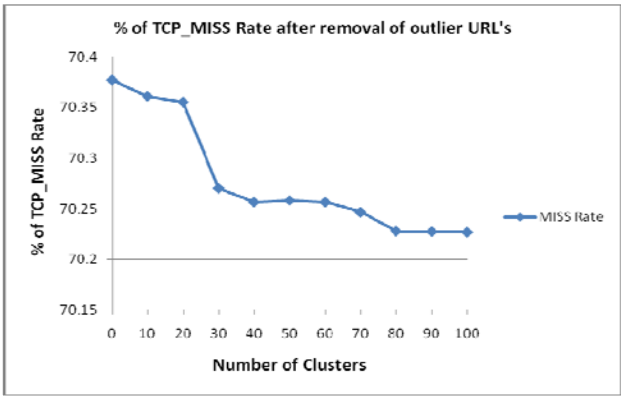


Fig 2. Graph between number of clusters and % of TCP_MISS rate.

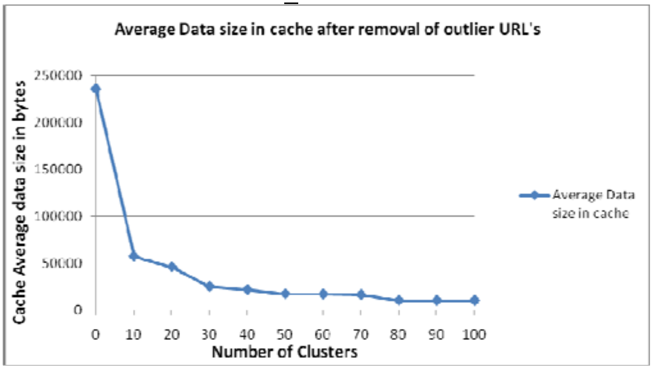


Fig 3. Graph between number of clusters and average data size retain in cache.

The graph plotted between number of clusters and average data size in bytes as shown in fig 3 also indicates that the average memory size needed to retain the cached web objects is reduced. Therefore more free memory is available to the newly cached web objects. It is seen that improvement in average cache data size is negligible beyond the number of clusters are chosen to be 80. Hence the maximum number of clusters is retained at 100 only.

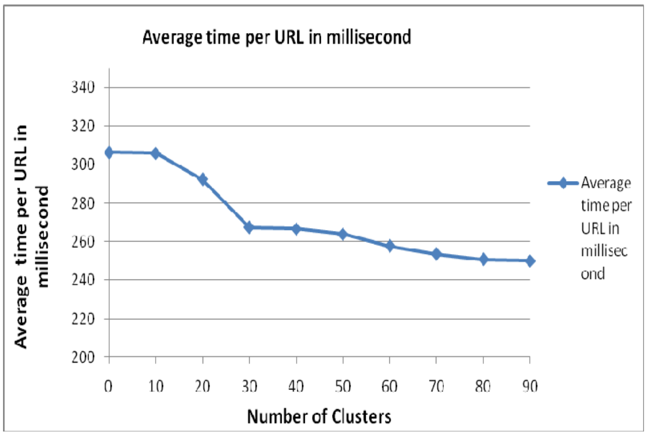


Fig. 4 Average time taken in millisecond to retrive URL from cache

Ratio between size of data (in bytes) and its retrival time (in millisecond) is known as average time (in millisecond).

$$\textit{Average Time} = \frac{\textit{Total time consumed (millisecond)}}{\textit{Total number of URL's requested in the same time}} \quad \text{....(2)}$$

It is seen , that average time is decreasing which indicates that we can retrieve more URL’s in less time. The average time per URL in ms for different number of clusters is shown in figure 4, which indicates that average time per URL decreases as number of clusters increase.

VII. CONCLUSION

If the web pages requested by the users are available in the cache of proxy server, the internet bandwidth will be saved, congestion on network will be reduced and also load on the original web server will also be reduced. This situation is represented by a hit of web page otherwise it will be a miss. Therefore our aim was to improve the percentage of hit rate and reduce the percentage of miss rate. To achieve this clustering method is applied to the available data to partition it different number of groups. Each time the entire group examined and found that the groups having small number of URLs are the outliers and removed from the memory of proxy server. After removal of these outliers the percentage of hit & miss and average data size is calculated. This process is repeated 10 times with the different number of clusters and each time a better result is found and percentage of hit rate increases, miss rate decreases, average memory size of cache decreases and average retrieval time per URL also decreases.

REFERENCES

[1] Sam Romano, Hala ElAarag, A Neural Network proxy cache replacement strategy and its implementation in squid proxy server, Neural Comput & Applic (2011) 20:59-78

[2] Jun Wang, Rui Min, Yingwu Zhu, Yiming Hu, UCFS- A Novel user-space, high performance, customized file system for web proxy servers, IEEE transaction on computers, Vol. 51, No. 9, September 2002.

[3] Ming-Kuan Liu, Fei-Yue Wang and Daniel Dajun Zeng, Web Caching : A Way to improve Web QoS, J. Comput. Sci. & Technol. , Mar 2004, Vol 19, No. 2

[4] Sai Rahul Reddy P : seminar report on Web Caching, school of Information Technology, Indian Institute of Technology – Kharagpur.

[5] Squid Web Proxy Wiki : http://wiki.squid-cache.org

[6] Evangelos P. Markatos, Dionisios N, Pnevmatikatos,Michail D. Flouris, Manolis G.H. Katevenis, Web-Conscious Storage Management for Web Proxies, IEEE / ACM transactions on networking Vol. 10 No. 6, December 2002.

[7] Ying Lu, Tarek F. Abdelzaher, Avneesh Saxena, Design, Implementation and Evaluation of Differentiated Caching Services, IEEE transaction on parallel and distributed systems Vol. 15 No. 5, May 2004.

[8] Waleed Ali Ahmed and Siti Mariyam Shamsuddin : Neuro-fuzzy system in partitioned client-side web cache, ELSEVIER Expert system with applications 38(2011) 14715-14725.

[9] Jake Cobb, Hala ElAarag : Web proxy cache replacement scheme based on back-propagation neural network, ELSEVIER systems and software 81(2008) 1539-1558.

[10] Moh. S. Obaidat and Humayun Khalid : Estimating Neural Network-Based Algorithm for Adaptive Cache Replacement, IEEE Transactions on System, Man, and Cybernetics-Part B : Cybernetics, Vol. 28 No. 4, August 1998.