INFORMATION TECHNOLOGY JOURNAL

Information Technology Journal 7 (2): 338-343, 2008
ISSN 1812-5638
© 2008 Asian Network for Scientific Information

Forming, Validation, Verification and Updation of Web Client Clusters Using Prefetching and Socket Clones Methods

A.M.J. Md. Zubair Rahman and P. Balasubramanie
Kongu Engineering College, Perundurai, Tamilnadu, India

Abstract: This study proposes a way of examining user interest based on the web pages accessed. Clustering these clients into interest groups so as to help in customized servicing and the service that has to be provided them is configured is discussed. The implementation aspects of web client clustering using prefetching and socket clone methods are presented. The utilization of various clusters is measured based on the number of accesses on the clusters by various clients.

Key words: Information retrieval, web clustering, customization

INTRODUCTION

With the web growing in leaps and bounds the numbers of applications on the web have increased many folds (Xiao and Zhang, 2001; Sit *et al*., 2002). The numbers of people using these applications also have increased many folds. Thus there has been an urge in people to use these applications. The increased load on servers thus has been a cry of the moment. Solutions have been sought after to provide for an easy solution to the increasing burden on these servers. Things have changed so much these days people have started to buy commodities that are personalized; web surfing thus need not be left behind too. When web surfing is personalized it brings into focus prefetching (Xiao and Zhang, 2001). Prefetching brings into question the ability of a particular server or a system to be able to predict the course of surfing meaning the web pages that will be accessed in the immediate future.

Xiao and Zhang (2001) proposes an approach for measuring similarity of interests among Web users, based on the interest items collected from Web user's access logs. A matrix-based algorithm is then developed to cluster Web users such that the users in the same cluster are closely related with respect to the similarity measure. As an application example, a Web document prefetching technique is proposed that utilises the similarity measure and clusters obtained. Experiments have been conducted and the results have shown that our clustering method is capable of clustering Web users with similar interests and the prefetching method is practical.

Sit *et al*. (2002) presents a new network support mechanism, called Socket Cloning (SC), in which an opened socket can be migrated efficiently between cluster nodes. With SC, the processing of HTTP requests can be moved to the node that has a cached copy of the requested document, thus bypassing any object transfer between peer servers. A prototype has been implemented and tests show that SC incurs less overhead than all the mentioned approaches. In trace-driven benchmark tests, our system outperforms these approaches by more than 30% with a cluster of twelve web server nodes.

Hence our approach is to cluster the web clients to provide some sort of customization. But basic challenge is how they should be clustered? What are the measures that can be used to cluster these users? This paper aims to provide a web clustering procedure for various clients which is an economical and makes web user comfortable while surfing. A matrix method (Xiao and Zhang, 2001) is used to cluster users based on these similarity scoring features. Users are compared against each other, their similarities measured, a matrix formulated. Then the users that express a significant amount of similarity are actually considered whilst the others dropped. The resulting matrix is a reduced form of the original matrix and contains only significant similar sets of users for a particular measuring feature. The matrix is then decomposed and put through the tests to eventually give the sets of users that have a similar kind of interest. The basic objective of client clustering can thus be defined as customization. Customization can only be achieved if they are targeted to a particular section of the browsing community. Specific pages can be retrieved for particular sections of the web surfing community which actually will translate into reduction of time and overhead costs when it is done before hand. Web pages can be prefetched (Xiao and Zhang, 2001) if the server is able to predict what to prefetch. This paper also discusses ways in which servers will be able to predict the pages that will be surfed in the

Corresponding Author: A.M.J. Md. Zubair Rahman, Kongu Engineering College, Perundurai, Tamilnadu, India

immediate future and can thus retrieve these pages and either store them remotely or can store them on board the system. The basic approach is in this study is to collect the user access details from the server logs and are analyzed to find out access patterns that are distinguishable. These patterns are then stored to give the server an idea of what to retrieve when the particular user is surfing the web. To cluster web clients into groups such that there will be an underlying level of similarity between them in terms of access areas. The clustered web clients are serviced using socket cloning method proposed by Sit *et al*. (2002). The purpose of undertaking such an endeavor is multifaceted, namely reduced load on servers, faster retrieval of data that is relevant, customization, etc. Thus a reduction in overhead costs is achieved.

SYSTEM MODEL

Given a set of m users U = $\{u_1, u_2, \ldots, u_m\}$ who access n web pages P = $\{p_1, p_2, \ldots, p_n\}$ of their subjective choices in any given time interval. There is some usage value associated with every user for a particular page denoted by use (p_i, u_j). The value of use(p_i, u_j) is 1 when page p_i is accessed by user u_j. The value of use(p_i, u_j) is 0 otherwise. The list of users is usually obtained from the system administrator.

Similarity measures

Type No. 1 similarity: Type No. 1 similarity is defined based on the total common pages accessed and is given by sim1.

Sim1(u_i, u_j) = Σ_k (use(p_k, u_i)* use (p_k,u_j))/√(Σ_k use (p_k, u_i)* Σ_k use (p_k,u_j)) where, Σk(use(p_k, u_i)) is the total number of pages that were accessed by the user u_i, Σ_k use (p_k,u_j) is the number of pages accessed by the user u_j. Σk(use(p_k, u_i)* use(p_k,u_j)) being the common number of pages accessed by the users u_i and u_j. The measure of similarity is based on the number of common pages accessed between 2 users.

Type No. 2 similarity: The type No. 2 similarity measure of similarity can be computed by counting the number of times these users actually access these common pages at all the sites visited. The similarity measure here is given by sim2.

sim2(u_i,u_j) = Σ_{ks}(acc$_s(p_k, u_i)$* (acc$_s(p_k, u_j)$)/(√(Σ_k Σ_s (acc$_s(p_k, u_i)$)^2* v(Σ_k Σ_s(acc$_s(p_k, u_j)$)^2)

where, acc$_s(p_k, u_i)$ is the total number of times user u_i accesses the page p_k at the web site s.

Type No. 3 similarity: Type No. 3 similarity is based on the most accurate measure of similarity would be the actual amount of time spent in viewing the pages involved. Then let t(p_k, u_j) be the amount of time user u_j spends on viewing page p_k. t = 0 if the page was not viewed at all, 1 if otherwise. Then type #3 is given by sim3

sim3(u_i, u_j) =Σ_k(t(p_k, u_i)* t(p_k, u_j))/((Σ_k(t(p_k, u_i))^2* Σ_k(t(p_k, u_j))^2).

Where, Σ_k(t(p_k, u_i))^2 is the square sum of the time the user u_i actually spent on viewing the pages in question. The denominator here is the inner product over time spent on actually viewing the common pages by users u_j and u_i. In this method of scoring even if two users access the same set of pages their similarity count might be less than 1 owing to the fact that the amount of time the user spent viewing these common pages differ to a large extent. This is justified by the fact that there are pages which are simply just used as a part of the navigation path that the user uses. These pages thus cannot be used as a criterion for measuring the similarity between the users in question. For credible similarity measuring pages that are actually viewed with some intent of analysis only be taken into account.

Type No. 4 similarity: The next and final measure of similarity is type No. 4 which involves the actual path of navigation that users u_j and u_i implement. The natural angle between the paths are calculated, it's actually the cosine between the 2 paths, i.e., cos($\infty\, {}_{Q}^{i}\, {}_{Q}^{j}$), where the final formula will contain the inner products of the paths involved over the feature spaces of the paths Q^i and Q^j. The similarity between web users is actually very much application dependant. In cases of subjective reasoning if a more complicated similarity measure is needed, an applicable similarity scoring feature must be defined for the said case.

When a navigation path is taken into context the links that are involved in the navigation path are primary. In the example discussed below the italicized words are the links in the web site that correspond to the connectivity graph under examination.

Six users are taken into consideration:

- **Main news:** 20 sec, National News: 15 sec, sport News: 43 sec, Stock index: 52 sec, News weather: 31 sec, business: 44 sec
- **Music box:** 11 sec, Stock *update*: 12 sec, Jungle Book: 13 sec, Books: 19 sec
- **Main news:** 33 sec, Movies Box :21 sec, Stock index: 44 sec, Sport News: 53 sec, Stock index:61 sec business:31 sec
- **Main news:** 19 sec, National News: 21 sec, Sport News: 38 sec, Stock index 61 sec, News business: 24 sec, Event News: 31 sec, News weather: 19 sec, Business: 39 sec
- **National news:** 32 sec, Stock index: 17 sec, News Jordan : 64 sec, Stock index: 19 sec, business: 50 sec

Table 1: Similarity statistic of six different users based on sim1 estimation

1	0.224	1	1	0.894	0.894
0.224	1	0.224	0.224	0.25	0.25
1	0.224	1	1	0.894	0.894
1	0.224	1	1	0.894	0.894
0.894	0.25	0.894	0.894	1	0.75
0.894	0.25	0.894	0.894	0.75	1

- **Main news:** 17 sec, Stock index: 33 sec, Sport News: 41 sec, Stock index 54 sec, business News: 56 sec, News: 47 sec

With the data given above for six different users a similarity matrix is computed which results in a mxn matrix called the similarity matrix. Using the first measure of similarity sim1 the similarity matrix SM is as follows:

Looking at the above given Matrix in Table 1 compare it to the log given before it, the first and the third users have similar patterns of visitation hence the measure of similarity between them is 1. While the access paths of the first and the second users are not the same so the measure of similarity is 0.224. Between the first and the fifth users the degree of similarity is given as 0.894 which naturally indicates a higher degree of similarity between the users than in the case between the first and the second.

Matrices for all measures of similarity are computed. The data is considered raw and has to be preprocessed to extract items of similarity and statistical significance from them. Clustering generally brings together items of similar interest into consideration with respect to a particular measure of similarity. A similarity threshold is thus determined to rid the matrix of any insignificance.

Notice in the above matrix that the leading diagonal is 1 and that the halves of the matrices on either side are the same. We use a threshold value of 0.7 to eliminate any insignificant relations that exist.

Permutation of the matrix: With insignificant elements in the matrices being eliminated the matrices now has clusters of users that have similar tastes with respect to a certain similarity measure. Thus the higher the value of SM (i, j) the closer the users u_j and u_i would be. The matrix after the computation will thus have elements with similar interests closer to each other. The next step will be to find out a dividing point that decomposes the said matrices. To permutate the given matrices a variable called the GA (Global Affinity) is first found out. The GA is an indicator of how closely users with similar interests are situated in the matrices. The GA value of a matrices is given by:

$$GA(SIM) = \sum_{i=1}^{n} A(i,i-1) + A(i,i+1)$$

Swapping of the rows is done to compute the GA, if found higher then rows are swapped to maintain relative positions of the columns.

Decomposition of the matrix: A dividing point in each matrix is to be calculated for the clustering of the users into 4 sub-matrices with similar kind of interests. The 4 parts will thus be called Upper left, Upper right, lower left and Lower right sub-matrices.

SERVICE MODES

These days cluster based web based servers are an impending necessity as a solution to high traffic web hosting. The solutions that are around suffer from the maladies that arise from the dispatcher based systems. These systems have a major drawback of single-point-of-failure. Because all the traffic is centered on one point in the network, these points in the network become crucial to the proper functioning of the network in general. To overcome this big disadvantage Distributed Packet Writing was invented which uses more than 1 dispatcher.

This method has its variations. The Layer-7 methodology implements the TCP splicing and TCP handoff. In TCP splicing the dispatcher (Sit *et al*., 2002; Cassalichio and Collajanni, 2001) itself is used as a proxy to reroute the HTTP response back to the client. In the TCP handoff methodology only the endpoint is passed to the resultant node so the server literally bypasses the dispatcher and directs the HTTP responses back to the client directly as shown in Fig. 1.

The network support mechanism that will actually aid the service to the clusters is called socket clone and will be discussed shortly. A socket clone is an opened socket which can be migrated between clusters.

Socket clone: A socket clone (Sit *et al*., 2002) is an opened socket which can be efficiently migrated between cluster nodes. Thus subjective to the clusters the needed processing is thus shifted to the cluster, mostly having some mechanism like a local cache for storage of prefetched documents.

The clone has three components as shown in Fig. 2:

- SC client
- SC server
- Packet router

The SC client provides a system call interface to the web server in the node. If the web server is unable to handle the request it deputizes another server to handle the request. It does so by using the system call given to

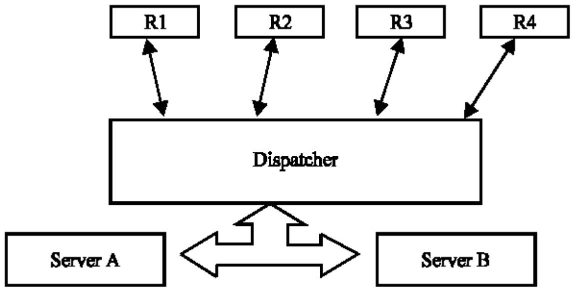

Fig. 1: TCP Splicing for four client requests (R1, R2, R3 and R4) with one dispatcher for two servers (A and B)

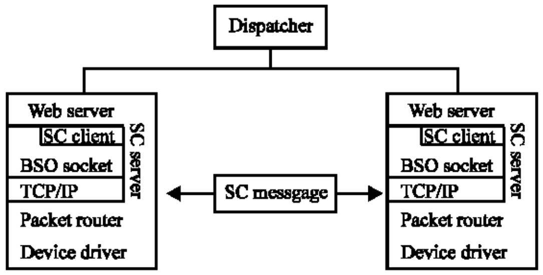

Fig. 2: Architecture socket clone

it to clone the socket. The SC client packs all the information of the opened socket and passes it on to the remote node through a persistent connection. This is called an SC message. When the new server that was deputized receives the message it creates a socket called a clone. The status is now worked backwards. The clone now becomes native to the new server and subsequent packets will go through its protocol stack. The output is generally sent to the Client. Usually after successful cloning the SC server will inform the packet router to route the packets to the clone's node. During the processing though packets from the client reach the original node and are then routed to the clone's node while HTTP responses are sent directly to the clone's node.

The clone and the original socket have the same states after cloning. The states change only after they have actually started sending out packets. As further processing of the requests from the client take place their status have to be synchronized which is done by a synchronizer which is probably a timer like device. A state of consistency has to be thus maintained to enable the original socket to process other requests further. To keep up the synchronization the status of the original clone is usually constantly updated to keep it in consistent form. It is usually done in the form of inter-nodal

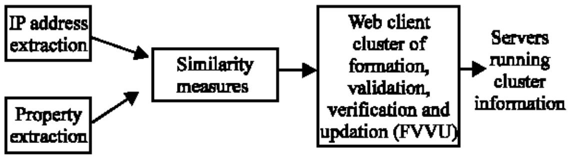
IP address extraction
Property extraction
Similarity measures
Web client cluster of formation, validation, verification and updation (FVVU)
Servers running cluster information

Fig. 3: Proposed system architecture

communication and rarely affects scalability or processing. Explicit synchronization is not needed: the router is in charge, it uses the acknowledgment packets received from the client to update the status of the original socket, updating the status and also aiding in continuing with the normal transaction. Thus a triangular mode of implementation is thus introduced.

PROPOSED ARCHITECTURE

The client interests are tracked using its IP address and major properties of the client choices are extracted as shown in Fig. 3. This raw information is used to form the web client clusters based on the similarity measurements which are proposed earlier. The most frequent items are determined and formed as a cluster. These clusters are under the continuous process of verification, validation and updation (FVVU). Now, the clients are served effectively by the servers by using this clustered information in a customized manner. TCP splicing methods and cloning methods are used to service various requests from clients.

RESULTS

Implementation of the clone in a web server: The server software in a cluster node first establishes connections with the clients directly and parses the request. A mapping function is used to decide on which node would process the request. The fact that the clustering of clients is already accomplished is to be remembered here. The mapping function itself can be of many types:

- Load based
- Location based
- Cache based

If a particular server is chosen to handle the request, it then handles it, else it simply clones the request and passes it onto the server that will actually service the request. For persistent HTTP requests an efficient and scalable mechanism for the cloning the socket multiple times is provided so that every request is served by the most appropriate server. In an event of a request being handled by a clone then a server in the clone's node will subsequently ignore the requests put to it. Thus it is now clear that clones are only meant for subjective processing and not used generally for routing or any general purpose.

The main server usually handles the request and may choose to clone the request or handle it itself. Pipelining of requests is also allowed where a browser sends for a request even before a complete response from the previous request is got. For error checking parity bits are counted. After receiving the final response from the server the router is usually directed to process the next request.

Hybridization: Let's now run through the entire length of the process before the final concept of hybridization is introduced. The system first makes clusters and uses some form of enumeration for identification. These clusters will have activation flags which will operate on the binary values of 0 and 1. One being indicative of the cluster being serviced and 0 on the contrary. If the value of the flag is 0 then no system resources are assigned. When the search criteria are given they are first checked for and then related documents are pre-loaded. Facilities for dynamically modifying the clusters of users are also included.

A router table like structure is maintained where cluster ids' and their usual access caches are addressed. So when a client logs on to the network the matching cluster of the client is found based on previous records and prefetching is done, however in the event of the user trying to access a page that is not usual to the personalized interest then the user is almost immediately assigned to a matching cluster if found and documents pertaining to the new cluster prefetched. In the event of simultaneous accesses common pages are cached and are referenced simultaneously for the different clusters that are being served. Time slicing is done when servicing simultaneous requests. Each time slot is split into equal parts based on the number of clients that are to be serviced.

Performance analysis: It is observed from Table 2 that the performance of the proposed is increased with cloning of the client requests by the dispatcher to appropriate server. The performance of the system is good when TCP splicing is done on various servers that are running clusters. The utilization is good for various requests of web pages and the same is shown in Table 2 and 3. It is to be noted that the clustering is more successful when there is similarity between web client requests. Table 4 shows the performance of the servers that are running various cluster data. It is observed that the clustering is more successful because every cluster is utilized more than 90% and it is indication good clustering. The FVVU factor also affects the performance of the proposed system.

Table 2: Performance of the system with and without splicing and cloning

File	Utilization (%)		
size (MB)	Without splicing	With TCP splicing	With clone
100	25.60	82.12	95.10
200	24.70	79.02	94.54
300	22.12	78.32	97.41
400	21.21	76.43	92.11
500	22.12	70.10	91.21

Table 3: Performance of the system with and without clusters

	Utilization (%)	
No. of clients	Without clusters	With clusters
100	25.60	95.10
200	24.70	94.54
300	22.12	97.41
400	21.21	92.11
500	22.12	91.21

Table 4: Load on the servers that are running various clusters

Cluster name	Utilization (%)
1	94.5
2	93.7
3	97.8
4	95.6

CONCLUSIONS

This study has thus devised a robust and dynamic plan for clustering web clients. The implementation of the socket clone along with the matrix based method of classifying clients is a novel idea that has proved efficient by its performance. The elimination of unwanted clients from clusters has also proved to be a good plan on boosting the performance of the system as a whole. With systems like this in place safer, faster and more customized surfing of the web is to be expected in the not too distant future. However, the key performance factor of the system lies with Formation, Verification, Validation and Updation (FVVU) of the clusters.

Future enhancements: Being the age of virtual computing it is more appropriate if future enhancements have at least in part have something to do with virtual computing. Virtual servers to service separate clusters is a distinct possibility. Improved caching is another paradigm in consideration. Heuristic approaches towards clustering of clients are another approach. The complete elimination of dispatchers is a definite line of thought and instead has intelligent agents roam the network to absorb patterns and trends. These systems are complex and hard to implement. These kinds of systems will generally have a higher rate of accuracy than the existing systems.

REFERENCES

Cassalichio, E. and M. Collajanni, 2001. A Client Aware Dispatching Algorithm for Web Clusters Providing Multiple Services. http://www10.org/cdrom/papers/434/colajanni_html.htm.

Sit, Y., C. Wang and F. Lau, 2002. Socket cloning for cluster based web servers. IEEE International Conference on Cluster Computing, pp: 333-340.

Xiao, J. and Y. Zhang, 2001. Measuring similarity of interests for clustering web users. Database Conference. ADC 2001. Proceedings. 12th Australasian, pp: 107-114.