GeoIP Clustering: Solving Replica Server Placement problem in Content Delivery Networks by clustering users according to their physical locations

Seyed Jalal Jafari
Department of Electrical and Computer Engineering
Graduate University of Advanced Technology
Kerman, Iran
sj.jafari@gmail.com

HamidReza Naji
Department of Electrical and Computer Engineering
Graduate University of Advanced Technology
Kerman, Iran
hamidnaji@ieee.org

Abstract— **Content delivery networks have emerged to conquer massive requests problem in popular web services. One of the major issues in CDNs is the replica server placement problem. If the replica servers are geographically closer to the users, overall bandwidth consumption and the users' perceived latency will be reduced significantly. In this paper we have used IP geolocation services to achieve the users' locations in terms of latitude and longitude. Then we have clustered users according to their coordinates. Locating servers around the cluster centers, provides better performance of CDN.**

Keywords-content delivery networks; cdn; replica server placement; clustering; performance

I. INTRODUCTION

The rapid growth of the Internet in the last decade has intrigued plenty of users to include the Internet in their lifestyle. Consequently new types of contents has been published in the virtual environment e.g. videos and audio files. The Web growth has transformed communications and business services such that speed, accuracy, and availability of network-delivered content has become absolutely critical [1]. Media sharing and social networks have become so popular recently. The well-known Youtube video sharing service has over 800 million unique users each month. 72 hours of video are uploaded to Youtube every minute [15]. Facebook which is a social network has more than 500 million active users. Facebook responds to over than 1 million HTTP Request every minute. Obviously the traditional Client-Server system is not able to answer requests in this huge scale. Content Delivery Networks has come into existence to solve this issue.

Content Delivery Network (CDN), is a trusted high-speed overlay network which is used to deliver Web objects, static data and streaming multimedia content [2]. The main idea behind CDNs is to place web objects as close as possible to the end-users. A typical CDN is composed of three components:

1) Origin Server 2) Surrogate Servers[1]

3) Users

[1] Replica Servers, Edge Servers

Contents are initially created at the Origin Server. Then using some replication (or caching) policies the existing contents are replicated on the Surrogate Servers which are distributed across the globe. When a user requests for some content, the surrogate server which is geographically closer to that user will respond to her. If the surrogate server has the requested object in its local cache, it will respond to the user directly, otherwise it will either pull the content from other servers or redirect the user to the origin server. Since the location of the surrogate servers plays an important role in the users' perceived quality, replica server placement has become an important issue in CDNs [3]. If the surrogate servers are placed in an appropriate location, the massive contents don't need to travel the entire path from one side of the network to the other side. Therefore the overall bandwidth consumption will be decreased. Additionally users' perceived latency will drop significantly.

In this paper we propose a new scheme for the replica server placement problem. We use IP Geolocation services to get the latitude and longitude of the users. Then we cluster the users based on their geographical locations. It is obvious that placing the surrogate servers around the cluster centers results in better performance of CDN. Section II introduces Content Delivery Networks. In section III the Replica Server Placement problem has been discussed. Section IV explains the proposed scheme and the execution methods and results are explained in section V.

II. CONTENT DELIVERY NETWORKS

Content Delivery Networks has come into existence to solve the massive requests and heavy loads issues of popular web services (not to be mistaken with webservice APIs). A CDN is a collection of network elements arranged for more effective delivery of content to end-users [4]. The main idea behind CDNs is to move content to the edge of the Internet, closer to the end-users. When a client requests for an object the surrogate server which is closer to that client will respond to her. Figure 1 depicts typical architecture of a Content Delivery Network [5].

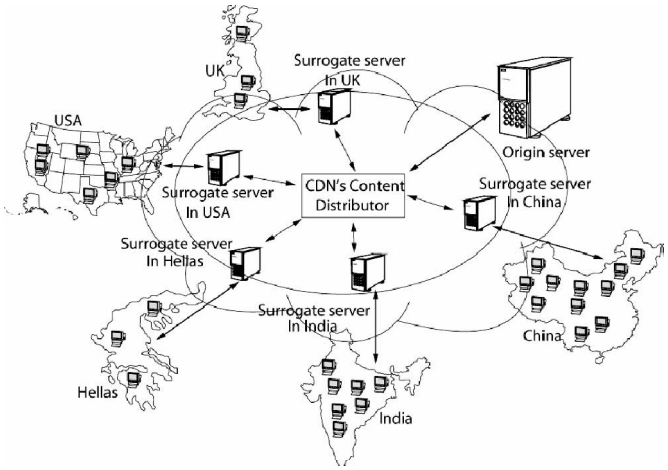

Figure 1. Typical architecture of a Content Delivery Network

For a popular web service there are plenty of clients accessing that service from different locations. Content Providers put the initial content on the origin server. Then CDN uses Content Replication or caching policies to move contents on the Surrogate Servers which are placed on the edge of the Internet infrastructure. Users' requests are directed to the closest server using request routing techniques. There are different approaches for the request routing problem. Pan, J. et al [6] have presented an overview of DNS-based server selection methods in CDNs. A taxonomy of request routing techniques can be found in [3]. After the proper server is selected, it receives the request. If it contains the requested content, it will respond to the client directly, otherwise it pulls the content from origin server or redirects the client to a server which can respond to her.

III. REPLICA SERVER PLACEMENT

Selecting the proper location for replica servers is an important issue in Content Delivery Networks. Surrogate servers should be distributed around the world such that CDN can balance the users' requests between them. In other words trying to minimize the number of surrogate servers we should consume the servers' resources as much as possible. As a server is utilized more, it means it has been placed in the right place. Generally we need to minimize two factors:

- Average perceived latency by users
- Overall bandwidth consumption in the network

There are two approaches toward the server placement problem in CDNs: Theoretical approaches and heuristic approaches.

A. Theoretical approaches

Theoretical approaches such as minimum k-center problem and k-Hierarchically well-Separated Trees (k-HST) model the server placement problem as the center placement problem [5]. The center placement problem tries to find a proper location for a given number of centers by minimizing the maximum distance between a node and the nearest center. K-HST uses graph theory to solve the server placement problem [7, 8].

B. Heuristic approaches

Due to the computational complexity of Theoretical approaches, some heuristic methods have been presented in the literature. Krishnan et al [9] have presented some methods on the placement of caches in a general network. They present a greedy algorithm based on the cache hit ratio. Qiu et al [10] have used this Greedy method for the CDN Server Placement problem. The Greedy approach is as follows: suppose that M surrogate servers should be selected among N servers. The greedy algorithm computes the cost of selecting each server in each stage of the algorithm. The cost can be a mixture of overall bandwidth consumed, overall perceived latency, request hit ratio, etc. In each stage the server that minimizes the cost function is selected. The algorithm continues until a desired number of servers are selected. It has been said that the greedy algorithm is the closest solution to the optimal one [11].

The server placement problem can be viewed as a clustering problem. Qiu et al [10] clustered the users into 1000 groups according to their geographical locations. They assumed that there can be a server for each cluster. Then they performed the greedy algorithm on these clusters. Under this configuration, the greedy algorithm outperformed hotspot server placement and tree-based server placement approaches. The greedy algorithm's median performance was 1.1 – 1.5 of optimal solution.

IV. THE PROPOSED SCHEME

In the expanding Content Delivery Networks, when the number users and consequently the amount of requests rise, the existing resources get overwhelmed. So we may need to add new servers to the existing infrastructure. The clients send the requests from all over the world. Therefore selecting a proper location for this new server is a crucial issue. As the surrogate servers are closer to the crowded locations, the users' perceived response time is better. On the other hand if we have the users' access log to the CDN, we are able to tell how the users are distributed across the globe. If we find a stable access pattern of the CDN users, we are able to judge whether the current location of the existing surrogate servers is optimal or not. If the servers are located in the sparse areas, moving them to somewhere in which users are more consolidated, will result in better performance of CDN in terms of bandwidth consumption and users' perceived quality.

The main idea of this paper is using IP Geolocation services to achieve users' approximate geographical locations. Then we cluster users according to their latitude and longitude. It is obvious that placing the surrogate servers near the cluster centers will result in better quality of content delivery. Also huge contents don't need to travel long paths in the network. Therefore bandwidth consumption will be reduced. The overall procedure of the proposed scheme is shown in Figure 2.

We assume that the CDN servers have been accessed for a long enough time such that we can approximately determine how the users are scattered around the world. The users' access details are logged into a database. The next step is to extract the users' IP addresses from the logs. This is easily done by extracting the source field of each log record.

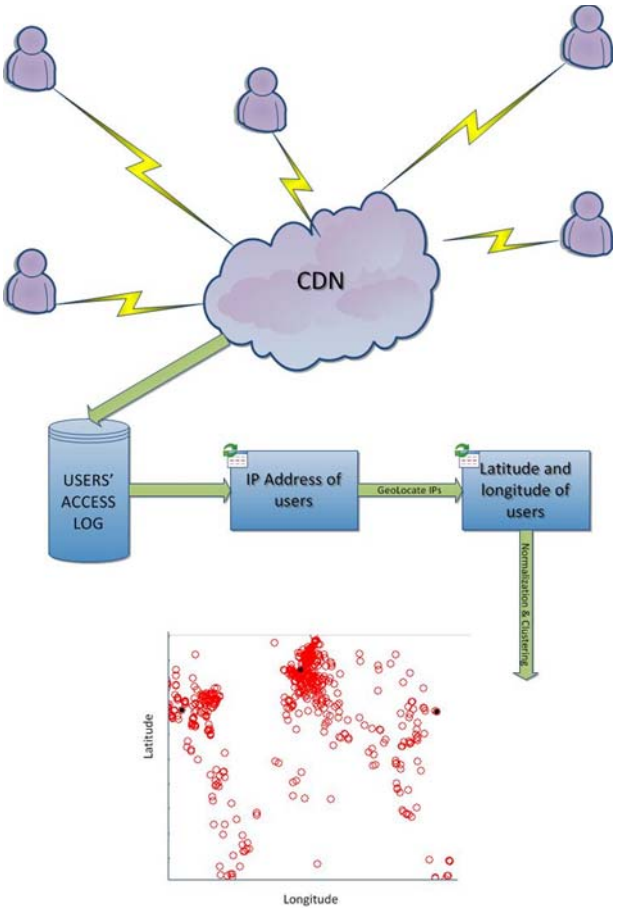

Figure 2. The procedure of placing surrogate servers

Since a user might have accessed the network more than one time, we perform a "distinct select" while extracting the IP addresses. Now having the list of distinct IP addresses we should turn them into geographical locations. The best representation of a geographical location is in terms of latitudes and longitudes. IP Geolocation services can take an IP address as input and return its geographical details including country, city, latitude, longitude, etc. in this experiment we just need the corresponding latitude and longitude for each IP address. After that we need to normalize the latitudes and longitudes and draw them on the Cartesian coordinates. Eventually we perform clustering algorithms on the obtained points.

A. *IP Geolocation Services*

IP Geolocation is a mechanism by which it is possible to get geographical location of a given IP address [12]. IP Geolocation providers store an updated list of IP ranges and their corresponding locations. For example IP Geolocation databases in the scale of country can be downloaded from maxmind.com for free. The detailed databases are for sale. The other service is freegeoip.net which provides some APIs which takes the IP address as input and returns geographical details of the given IP address in XML, JSON and CSV format. Figure 3 depicts an XML response for an IP Geolocation request.

Figure 3. The XML response for an IP Geolocation request

To automate the process of converting IP addresses into latitudes and longitudes, a Java application is designed which reads the IP addresses from a database and calls the IP Geolocation API shown in Figure 3 for each IP address. Then it receives the geographical details of each IP address in JSON format. Within the obtained JSON records we just need the latitudes and longitudes to determine a point on the earth for each user.

B. *Clustering users according to their coordinates*

Data clustering is a process of putting similar data into groups [13, 14]. In this study Subtractive Clustering and Fuzzy C-Means have been used to cluster the CDN users. The reason to select these methods among other clustering approaches lies in their different type of usages. In Fuzzy C-Means the number of clusters must be specified beforehand. Therefore Fuzzy C-Means is used when we are aware of the number of the Surrogate Servers. In a Subtractive Clustering the number of clusters can be variable. In fact it depends on the influence range we specify as the input of the algorithm. When the influence range is smaller, the radius of the produced clusters will be smaller too. So the number of found clusters will be increased.

In clustering algorithms the method of calculating distance between the cluster nodes can be important. Generally the distance between nodes is assumed equal to the Euclidean distance between them. Since the nodes we are working with represent latitudes and longitudes on the earth, the distance can be calculated more precisely using great-circle distance formulas. Haversine equation, which is specified in (1), is one of them. The Haversine formula takes latitudes and longitudes for two points on the earth as inputs and calculates the distance between them in kilometers. In this equation, φ is the latitude, λ is the longitude, and R represents the radius of the earth which is approximately equal to 6371 kilometers.

$$\begin{aligned} a &= \sin^2(\Delta\varphi/2) + \cos(\varphi_1).\cos(\varphi_2).\sin^2(\Delta\lambda/2) \\ c &= 2.\mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)}) \\ d &= R.c \end{aligned} \quad (1)$$

In order to achieve more accuracy in the clustering process, we can override the clustering algorithms in a way that they use Haversine formula to calculate the distance between nodes. But the Euclidian distance is a good approximation for our study. So for convenience we use the Euclidian distance.

V. EXPERIMENT EXECUTION AND THE RESULTS

In order to run and evaluate the proposed scheme we have used two datasets. In fact the datasets are originally two years of Apache access logs for the websites "googlecreeper.cjb.net" and "ohiovideoproduction.com". The former is a Swedish website which exposes Swedish users' Google searches and it has revealed its own access log too. The latter is a video production company in the US.

Using the proposed scheme, we have extracted the latitudes and longitudes of these datasets. Then the points were normalized in order to feed them to clustering algorithms. Figure 4 depicts the first dataset's distinct users. As it was mentioned before, this dataset belongs to a website in Sweden, so it is logical that most of its users are located in Europe. There are also considerable amount of users for this website in America. Figure 5 represents users of the second dataset. The website is an American video production company, so logically most of its users are located there. There are also some users from Europe and Asia too.

The Next step is to cluster these users according to their locations. As it was stated before, we use two methods for clustering: Subtractive clustering and Fuzzy C-Means Clustering. The former is for the time we don't know the number of the servers and we just want to evaluate how many servers is needed. While the latter is used when we have a fixed number of clusters and we are looking for a better place to put them.

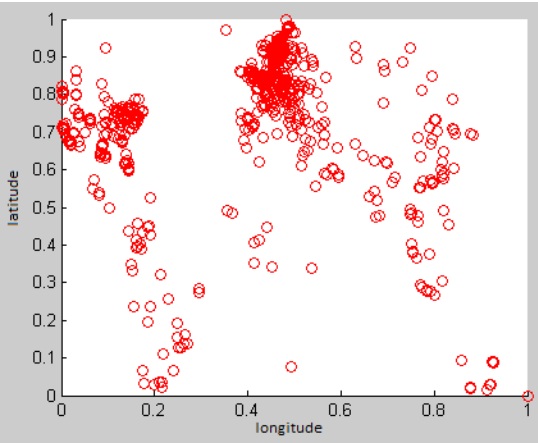

Figure 4. googlecreeper's user distribution

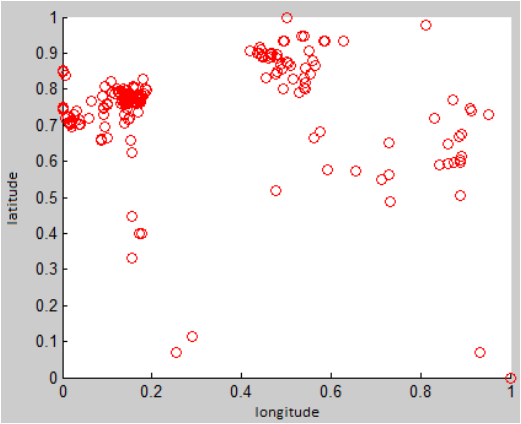

Figure 5. ohiovideoproduction's user distribution

In the following the result of clustering process is provided. We have executed the subtractive clustering with the influence range of 0.2 and FCM clustering with four predefined cluster numbers for both dataset.

Figure 6 is the result of subtractive clustering for "Google creeper" dataset which specifies three points as cluster centers. Figure 7 is the result of Fuzzy C-Means clustering for this dataset. The number of clusters has been determined equal to 4 for this experiment. The output of two clustering methods are almost the same, three major clusters has been detected. In the FCM method an extra cluster is detected in the densest part of the network.

Figure 8 and Figure 9 are the result of clustering algorithms for "Ohio video production" dataset. Subtractive clustering with the influence range of 0.2 results in two cluster centers. FCM with four predefined cluster centers produces two of the points in America section in which there are more users. As it is visible in Figure 8, Subtractive clustering could not find the cluster which lies in Europe. We can call this a flaw. But the FCM method covers for this issue and detects all the clusters. So it is better to aggregate the results of different clustering approaches at the end.

Eventually in order to convert the output cluster centers to a human understandable format, we should denormalize them. To do this we just have to put the points in the equation (2). The result is coordinates of the given cluster center. Having the latitude and longitude of the cluster centers, we know where is most suitable to place the surrogate servers of the CDN.

$$\begin{aligned} &\forall\ \text{NormalizedCenters}, X_{(\text{lat,lon})} \\ &C_{\text{lat}} = (X_{lat} \times (Max_{lat} - Min_{lat})) + Min_{lat} \\ &C_{\text{lon}} = (X_{lon} \times (Max_{lon} - Min_{lon})) + Min_{lon} \\ &Center\_Coordinates = (C_{lat}, C_{lon}) \end{aligned} \quad (2)$$

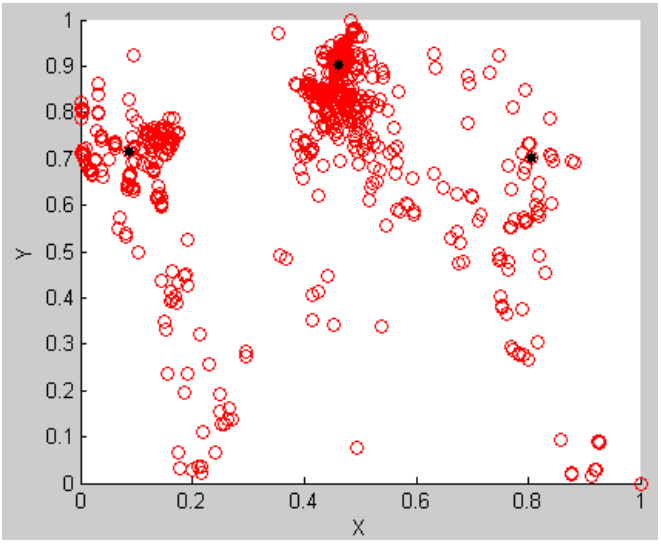

Figure 6. Subtractive clustering on googlecreeper, influence range=0.2

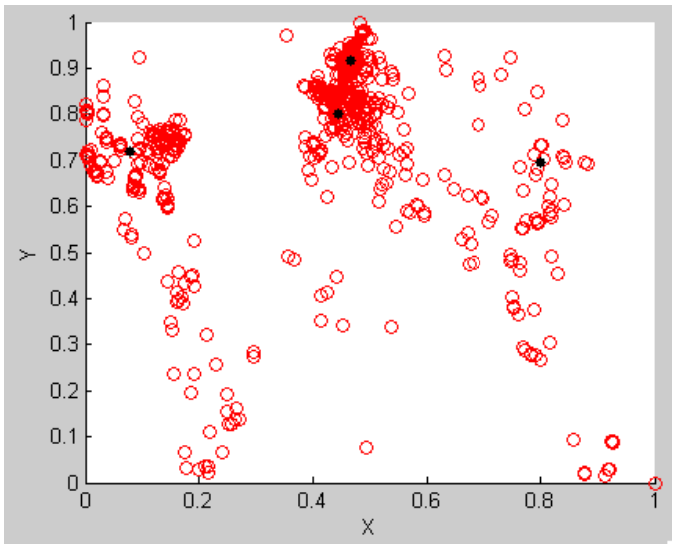

Figure 7. FCM clustering on googlecreeper, cluster num=4

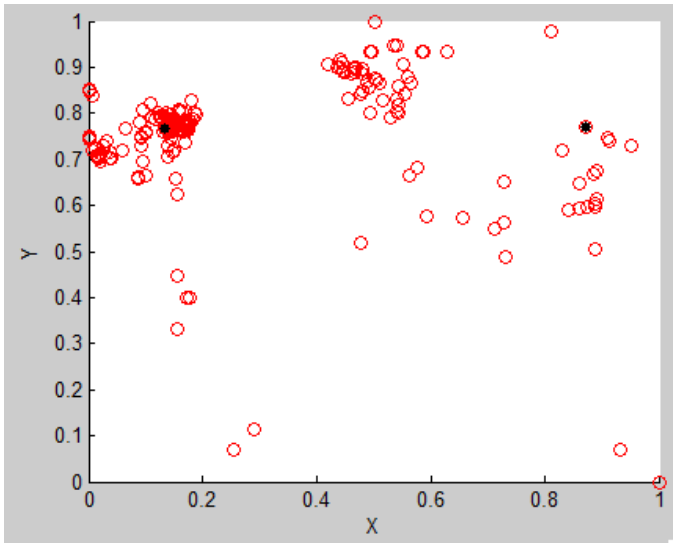

Figure 8. Subtractive clustering on ohiovideoproduction, influence range=0.2

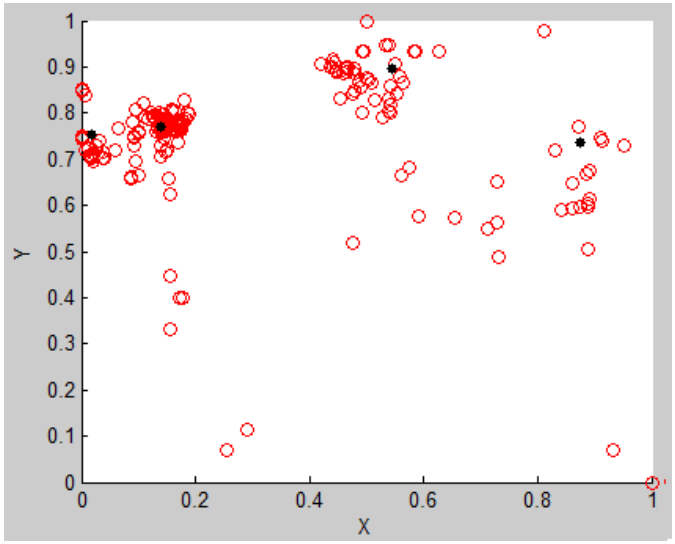

Figure 9. FCM clustering on ohiovideoproduction, cluster num=4

VI. CONCLUSION

Content Delivery Networks have emerged to improve the quality of service of popular web services. CDNs place surrogate servers on the edge of Internet infrastructure in order to deliver the contents to the users effectively. Finding a proper place for surrogate servers is an important issue in CDNs. In this paper we have used IP Geolocation services to find users' approximate locations on the earth in terms of latitudes and longitudes. Then we use subtractive clustering and Fuzzy C-Means clustering in order to put the users into similar groups. The final result of proposed scheme is the coordinates of appropriate locations to place the surrogate servers.

REFERENCES

[1] A. Vakali and G. Pallis, "Content delivery networks: Status and trends," *Internet Computing, IEEE,* vol. 7, pp. 68-74, 2003.

[2] K. Stamos, G. Pallis, C. Thomos, and A. Vakali, "A similarity based approach for integrated Web caching and content replication in CDNs," in *Database Engineering and Applications Symposium, 2006. IDEAS'06. 10th International*, 2006, pp. 239-242.

[3] A. M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report,* 2007.

[4] F. Douglis and M. F. Kaashoek, "Guest Editors' Introduction: Scalable Internet Services," *IEEE Internet Computing,* vol. 5, pp. 36-37, 2001.

[5] R. Buyya, M. Pathan, and A. Vakali, *Content delivery networks* vol. 9: Springer, 2008.

[6] J. Pan, Y. T. Hou, and B. Li, "An overview of DNS-based server selections in content distribution networks," *Computer Networks,* vol. 43, pp. 695-711, 2003.

[7] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "On the placement of internet instrumentation," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, pp. 295-304.

[8] Y. Bartal, "Probabilistic approximation of metric spaces and its algorithmic applications," in *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, 1996, pp. 184-193.

[9] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Transactions on Networking (TON),* vol. 8, pp. 568-582, 2000.

[10] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the placement of web server replicas," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2001, pp. 1587-1596.

[11] G. Peng, "CDN: Content distribution network," *arXiv preprint cs/0411069,* 2004.

[12] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards IP geolocation using delay and topology measurements," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 71-84.

[13] K. Hammouda and F. Karray, "A comparative study of data clustering techniques," *Tools of intelligent systems design. In Course Project, SYDE,* vol. 625, 2000.

14] M. Verma, M. Srivastava, N. Chack, A. K. Diswar, and N. Gupta, "A Comparative Study of Various Clustering Algorithms in Data Mining," *International Journal of Engineering Research and Applications (IJERA),* vol. 2, pp. 1379-1384, 2012.

[15] http://www.youtube.com/t/press_statisti