International Journal of Grid and Distributed Computing
Vol. 10, No. 9 (2017), pp.1-12
http://dx.doi.org/10.14257/ijgdc.2017.10.9.01

Edge Server Selection in Distributed Content Delivery Network using K-means Square Classification Algorithm and Ant Colony Optimization

Debabrata Sarddar[1], Sougata Chakraborty[2]* and Priyajit Sen[3]

[1]Assistant Professor, Department of Computer Science & Engineering, University of Kalyani, West Bengal, India
[2]Senior System Engineer at IBM India Private Limited, Kolkata, West Bengal, India
[3]Student of Master of Technology, Department of Computer Science & Engineering, University of Kalyani, West Bengal, India
[1]dsarddar1@gmail.com, [2]me.sougata.chakraborty@gmail.com, [3]priyajit91@gmail.com

Abstract

The request-routing in a Distributed Content Delivery Network directs users to the closest edge server that can serve the request best depending upon the metrics such as network proximity, client perceived latency, distance, and replica server load. Edge servers provide efficient access to the request users depending on their best availability as per the smallest distance and free space in the storage. DCDN routing is exploited by an organization where several edge servers are connected to a service node. The service node keeps information about each edge server connected to it and attempts to implement a global load balancing policy. Today's fastest world faces the problem of data usage and delivery of the multimedia contents due to the overburden of multiple requests to the web server over the course of daily business hours. Thus, to minimize the problem over data access it is required to use a fastest algorithm to search the edge servers from the request users. In this paper, we have used K-Means Clustering approach and Ant Colony optimization to minimize the problem by forming clusters as per the distance from the request users and cluster head (Centroid) and finally optimize the problem.

***Keywords**: K-means square algorithm, Ant colony optimization, Distributed content delivery network, Request routing, Inter-cluster edge server.*

1. Introduction

Content Delivery Networks (CDNs) [1], [2] offer fast and reliable content delivery and reduce the communication bandwidth by caching and replication. A CDN [7] receives the content from an Origin server, and then replicates it to its Edge cache servers; the content is sent to an end-user from the "closest" Edge server. CDNs [8] are designed to support scalability, to increase reliability and performance, and to provide better security. A CDN [2] replicates the content from the origin server to cache servers, scattered over the globe to deliver content to end-users in a reliable and timely manner. Content distribution on the internet is a considerable research attention. It is the combination of high-end computing technologies with the highest performing networking infrastructure and distributed replica management techniques.

Received (March 22, 2017), Review Result (June 27, 2017), Accepted (July 10, 2017)

ISSN: 2005-4262 IJGDC

Figure 1. Content Delivery Network

1.1. Content Delivery Network (CDN)

A content delivery network is distributed system of servers spread over the globe. The edge server serves the requests made by the request users. Request users request for the web contents in content delivery network. Contents are redirected by the CDN from original server to the closest edge server of a request user.

The benefit of the modern CDN is the multiple copies of the web content are resided across the web. Thus, distributing of that content across the Internet removes the inevitability for customer requests traversing many routers to directly access the facilities of the web publisher. This reduces traffic routed through the Internet as well as the delays associated with the routing of traffic. Thus, a CDN can be expected to reduce latency between client and server.

The shortfall is it would be impractical and quite next to impossible for almost all organizations to maintain identical servers deliberately positioned around the world. Thus, most of the organizations need to rely on the third party CDN operators and this includes the questions of cost and support. There is also the need to ensure the CDN provider can update the web publisher's server in a timely manner.

1.2. Distributed Content Delivery Network (DCDN) Architecture

DCDN architecture [3] [8] will be a hybrid architecture which integrates some of the major features of conventional client-server CDN and an academic peer-to-peer CDN.

A set of Local DCDN Servers and countless DCDN Surrogates are networked together to deliver requested Web content to the clients. The main components of DCDN architecture are Content providers, DCDN servers and DCDN surrogates that are arranged in a hierarchical order.

Content Provider: It is that entity that request to distribute its Web content through DCDN.

DCDN Administrator: The entire DCDN network is managed, supported and run by a team of administrators.

DCDN Servers: DCDN servers are basically acting as re-directors that will only have the knowledge about the location of the content. It may function as a buffer system, which help to push the content provided by the content providers to DCDN surrogates.

DCDN Surrogates: DCDN surrogates [3] are the large number of Web users who offers resources in terms of storage capacity, bandwidth and processing power to store and make available DCDN Web content.

DCDN Client: The clients are the end users who make a request for a Web content using a Web browser.

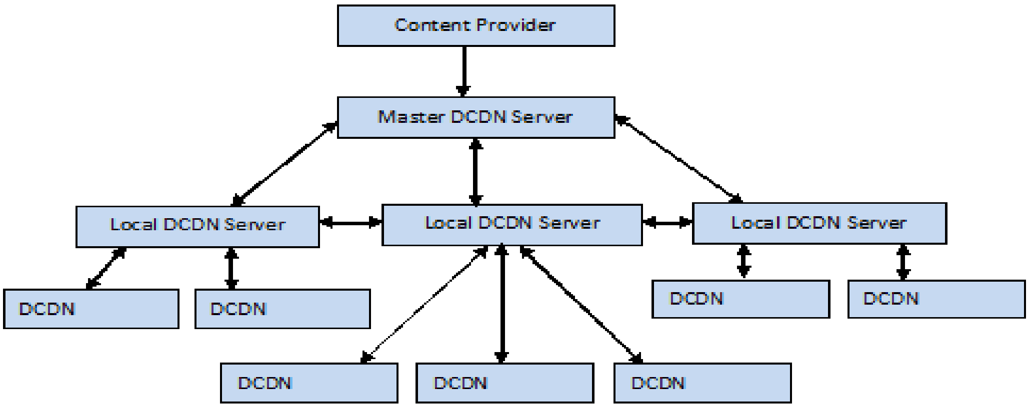

Figure 2. DCDN Content Distribution Architecture

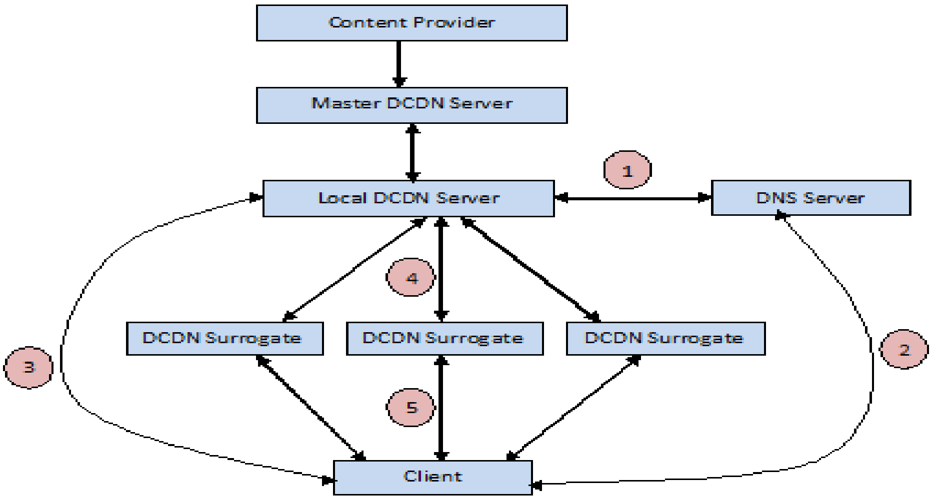

Figure 3. DCDN Content Delivery Model

1.3. K-Means Square Classification Relating to the Edge Server Selection

K-means is one of the simplest unsupervised learning algorithms to solve the clustering problem [12]. The procedure follows a simple and easy way to classify a given set of data through a certain number of clusters (assume k clusters). The main idea is to define k edge server, one for each cluster (the set of request users). These edge servers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. Now consider each point belonging to a given data set (from the request users) and associate it to the nearest edge server.

Given a set of request from the request users ($x_1, x_2, \ldots, x_n$), where each observation is a d-dimensional real vector, k-means clustering aims to severance the n observations into $k\,(\le n)$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (sum of distance functions of each point in the cluster to the k center). That is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \qquad (1)$$

Where μ_i is the mean of points in S_i.

1.4. Ant Colony Optimization

It is a probabilistic technique. It helps ants to search the optimal path in the graph based on their behavior between their colony [9] and source of food. Ants navigate from nest to food source. Ants are blind; therefore, shortest path is discovered via pheromone trails. Each ant moves at random. Pheromone is deposited on the path. More pheromone on path increases probability of path being followed.

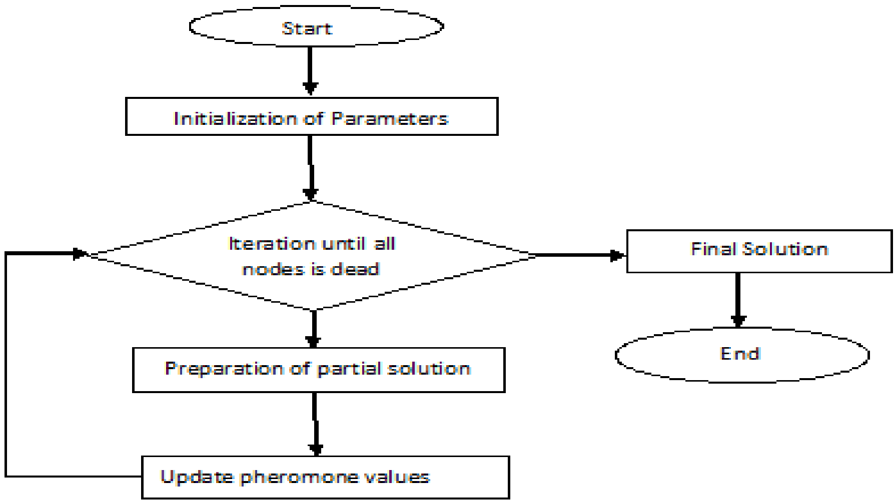

Figure 4. Flowchart Showing the Ant Colony Optimization Technique

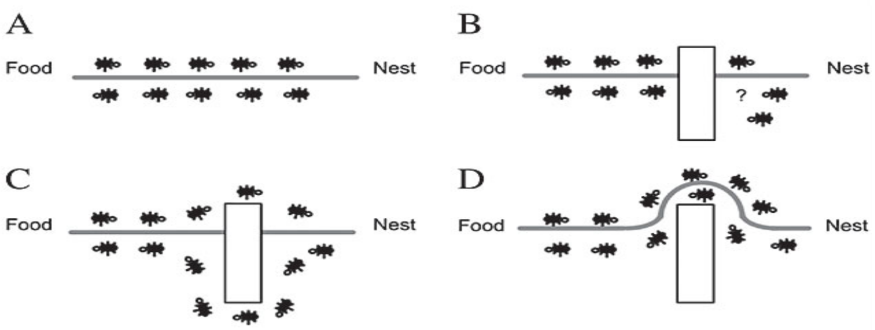

Figure 5. The Ant Colony Optimization Process Showing Movement of Ants to Search Food from Nest

Two ants start from their nest (left) and search for the shortest path to a food source (right). Initially, no pheromone is present on either trail, so there is the same chance of choosing either of the two possible paths. Suppose two different ants choose the upper trail and the lower trail. The ant that will choose the upper (shorter) trail will return faster to the nest. Thus, there is a greater amount of pheromone on the upper trail as on the lower one. The amount of pheromone discharged on a path is inversely proportional to the distance to the food source using that path. Therefore, the shortest path to the source of food will get maximum pheromone deposited on it. The probability of choosing the upper (shorter) trail the next ant will be higher. More ants will choose this trail, until all (majority) ants will follow the shorter path.

2. Related Work

In the paper "An Efficient Edge Server Selection in Content Delivery Network using Dijkstra's Shortest Path Routing Algorithm with Euclidean Distance" by Sougata Chakraborty and Debabrata Sarddar (2015), we have seen how Dijkstra's Shortest Path Routing Algorithm with Euclidean Distance have been used for finding the shortest distance between the request users and the edge server.

In their paper "An overview of DNS-based server selections in content distribution networks", Sarddar et al. (2014) have proposed Domain Name Server based edge server selection algorithm in CDN network.

In the paper "An Architecture for Distributed Content Delivery Network" (2007) Jaison Paul Mulerikkal has proposed the Distributed Content Distribution Architecture and delivery models.

In the paper "Optimization of Cloud Database Route Scheduling Based on Combination of Genetic Algorithm and Ant Colony Algorithm", Zhang Yan-huaa et al. (2011), we have seen the genetic algorithm and the ant colony optimization algorithm are used together for optimization of cloud database as well as route scheduling.

In the paper "Ant colony optimization: Introduction and recent trends" by Christian Blum (2005), we have read vividly on Ant Colony Optimization and its recent trends.

In their paper "Using Surrogate Servers for Content Delivery Network Infrastructure with Guaranteed QoS", K. H. Wong *et al.*, (2013) have proposed a guaranteed Quality of Service technique using Surrogate Servers for Content Delivery Network Infrastructure.

In their paper "Globally Distributed Content Delivery", J. Parikh *et al.*, (2002) have proposed a globally Distributed Content Delivery Network.

In their paper "A Greedy Ant Colony Forwarding Algorithm for Named Data Networking", Chengming LI *et al.*, have proposed a greedy Ant Colony Forwarding Algorithm.

In their paper "On the use and performance of content distribution networks", B. Krishnamurthy *et al.*, have proposed the use and performance of the content delivery network.

In their paper "Centralized scheduler for content delivery network", Jun Li et al. have proposed the Centralized Scheduler Concept for the content delivery network that keeps track of all the request and service provided in the network.

In their paper "An efficient k-means clustering algorithm", Khaled Alsabti et al. have proposed efficient K-Means clustering algorithm for to enhance the performance of a cloud network.

In the paper "Optimal Path Identification using Ant Colony Optimization in Wireless Sensor Network" by Aniket. A. Gurav and Manisha. J. Nene, we have found the ACO is used for optimal path identification and the experimental pheromone values in a tabular method.

3. Proposed Work

Our proposed approach is to find and allocate the edge server that is nearest to the request users and storage is best available for the allocation. We have used K-means square clustering method and ant colony optimization technique to route the edge server to the request user and thus it will form different cluster of edge servers for different request users.

Algorithm of K Means Square Classification:

Step1: Select k out of the given n request as the initial cluster centres. Assign each of the remaining $n-k$ request mappings to one of the k clusters; a pattern is assigned to its closest edge server/cluster.
Step 2: Re-compute the cluster centres based on the current assignment of requests.
Step 3: Assign each of the n request to its closest edge server.
Step 4: If there is no change in the assignment of request successive iterations, then stop; else, go to Step 2.

Algorithm of Ant Colony Optimization:

Step 1: An edge server starts calculating shortest path and selects the destination node from all other source nodes *i.e.*, an ant is created and scans the graph G . The goal of this scan is to find the optimal solution.
Step 2: Every ant has its own memory used for storing information about travelled path. This memory can also serve to evaluate of the solution.
Step 3: The execution begins at state x_s^k and it has one or more ending constraints e^k . Let the actual state of an ant is the state x_r = (x_r -1, i) and no ending constraints is complied, so the ant moves to node j in the neighborhood of the state $N^k(x_r)$ and the ant moves to the new state (x_r , j) $\in X$.
Step 4: The next ant execution depends on the probability and that is calculated based on the pheromone quantity. It also considers its local memory and the acceptance of this step.
Step 5. If the ant goes to new component of graph G_c , it updates the value of corresponding pheromone.
Step 6. The ant can update pheromone values after the construction of reverse path by editing associate pheromone values.

The above process is repeated for many times and thus an edge server comes to know about the optimal path between it and the destination nodes or other edge servers and it can calculate distance between the source node and all other destination nodes.

Hence, we can simply find out that which node could be the nearest neighbor edge server of any requested user and which optimal path could be used for forwarding and routing the multimedia contents via one or many edge servers.

4. Simulation Result

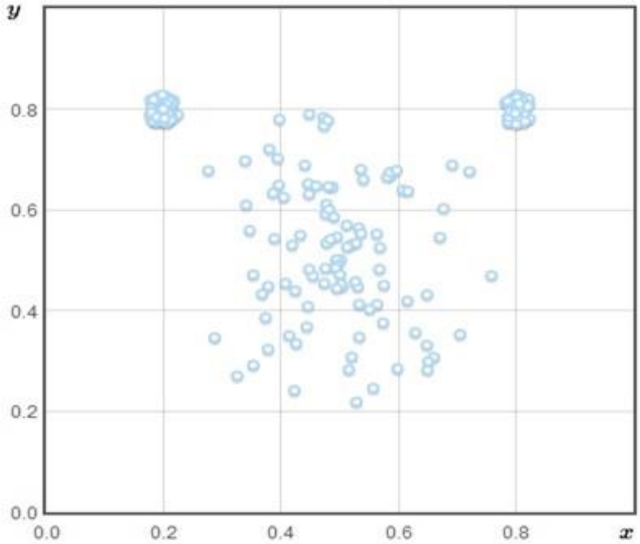

Figure 6. Randomly Generated Data (Request Users)

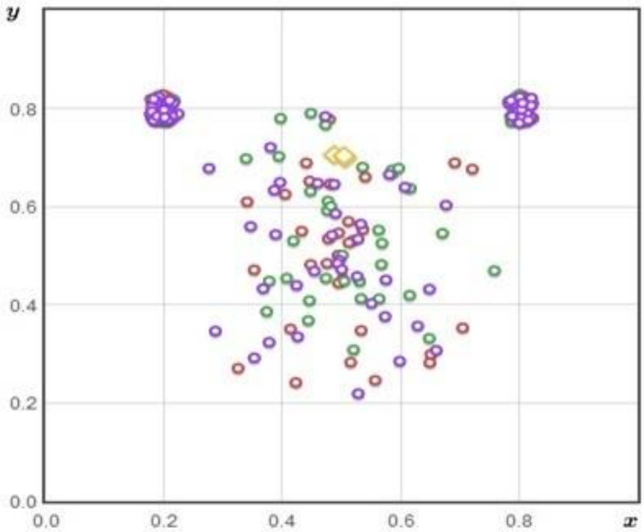

Figure 7. Initialization of Cluster

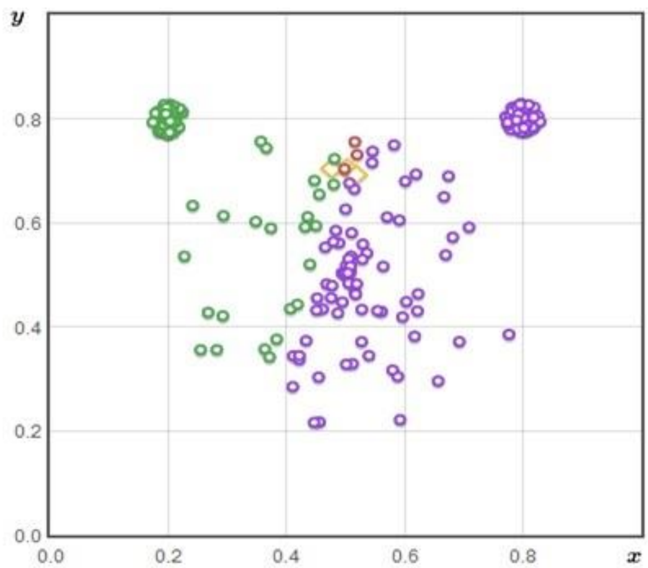

Figure 8. Assign Data Points to the Closer Cluster

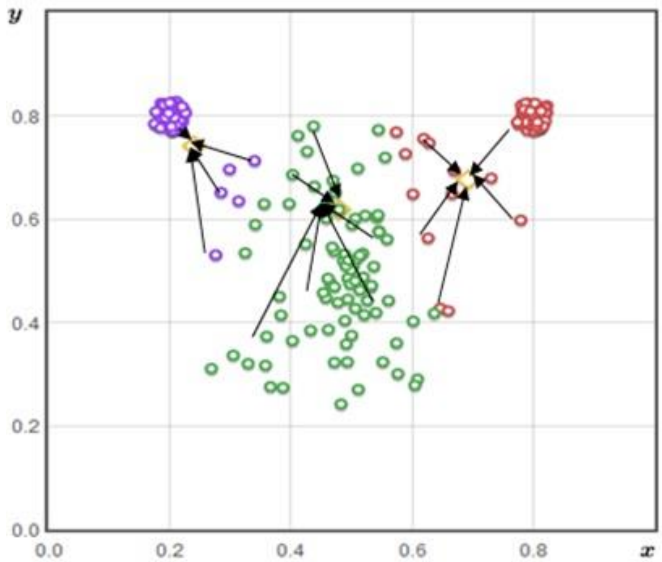

Figure 9. Calculate Center of Each Cluster

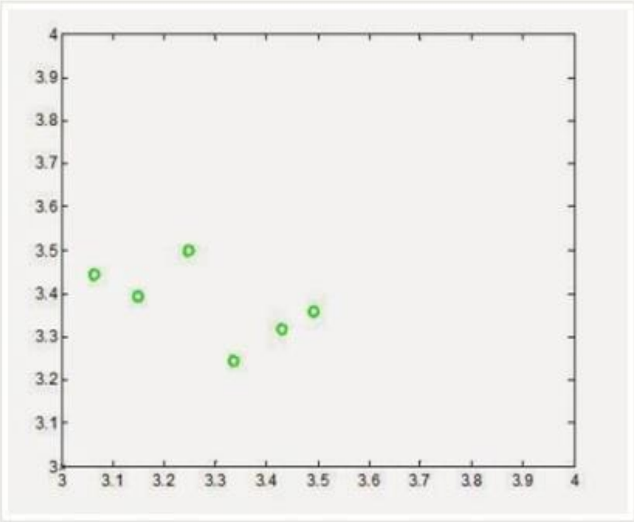

Figure 10. Number of Ants Move in Different Directions

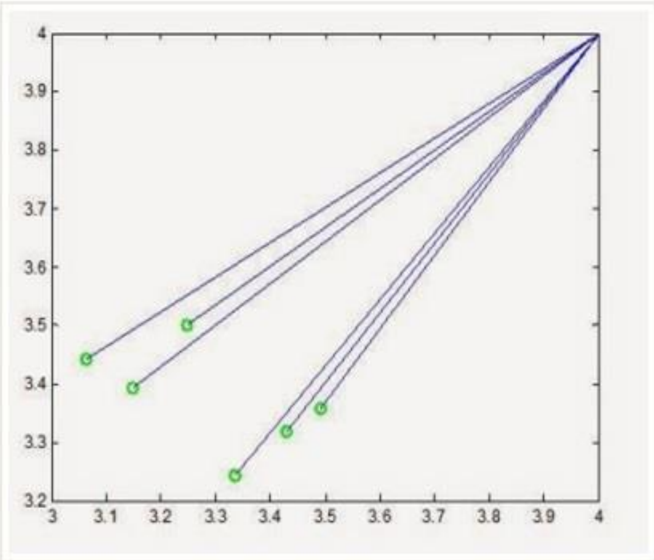

Figure 11. Distance of Each of these New Points with Destination

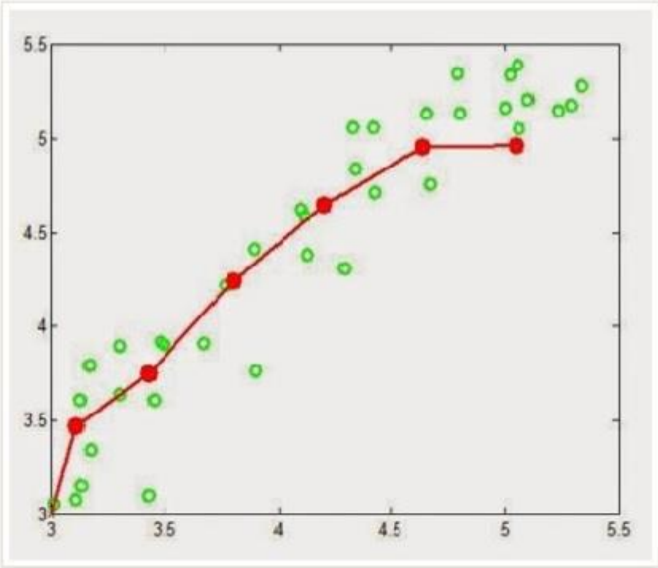

Figure 12. Closer to the Optimal Path Found

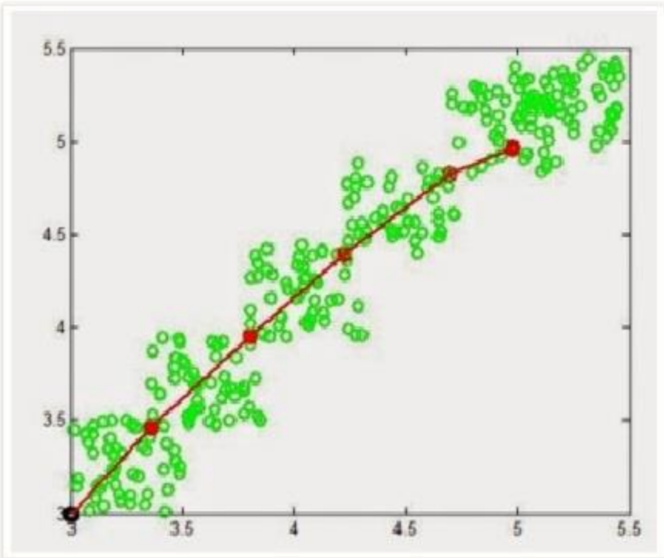

Figure 13. Optimal Path Found based on a Prior Experience

5. Conclusion

We have proposed an approach for the selection of edge servers so that the web users or the cloud users might get hassle free faster service during browsing over internet and downloading multimedia contents from any popular website if the contents are coming from the nearest neighbor edge server and traverse along the shortest path by ACO algorithm. Thus, the performance of the network is increased and many people can get the access at the same time even in daily business hours. The beneficial aspect of our research work is whenever the number of requests in the DCDN will increase, the server will never go down as it forms clusters and the request users can find the best edge server as a cluster head efficiently. In our future work, we can plan to develop a new provisioning algorithm which might take care of the scenario like if the edge server crashes or any technical failure occurs. In those cases, the information of any edge server will also be available in the inter-cluster edge servers and a smarter backup and recovery technique will be introduced.

References

[1] S. Chakraborty, D. Sarddar, "An Efficient Edge Server Selection in Content Delivery Network using Dijkstra's Shortest Path Routing Algorithm with Euclidean Distance", International Journal of Computer Applications (0975 – 8887), vol. 117, no. 4, **(2015)** May.

[2] J. Pan, "An overview of DNS-based server selections in content distribution networks", Elsevier Computer Networks, vol. 43, **(2003)**, pp. 695-711.

[3] J. Paul Mulerikkal, "An Architecture for Distributed Content Delivery Network", M.S. thesis, Distributed Systems and Networking, School of Computer Science, Royal Melbourne Institute of Technology University, Melbourne, Australia, **(2007)** July 17.

[4] Z. Yan-huaa, F. Leia and Y. Zhia, "Optimization of Cloud Database Route Scheduling Based on Combination of Genetic Algorithm and Ant Colony Algorithm", Procedia Engineering, Elsevier Ltd. Open access, vol. 15, **(2011)**, pp.3341-3345.

[5] C. Blum, "Ant colony optimization: Introduction and recent trends", Elsevier Physics of Life Reviews, vol. 2, **(2005)**, pp. 353-373.

[6] H. Al-Zurba, T. Landolsi, M. Hassan and F. Abdelaziz, "On the Suitability of using Ant Colony Optimization for Routing Multimedia Content Over Wireless Sensor Networks", International journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC), vol. 3, no. 2, **(2011)** June.

[7] K. H. Wong, P. Y. Lau and S. Park, "Using Surrogate Servers for Content Delivery Network Infrastructure with Guaranteed QoS", Journal of Advances in Computer Networks, vol. 1, no. 1, **(2013)** March.

[8] J. Parikh, H. Prokop, R. Sitaraman, J. Dilley, B. Maggs and B. Weihl, "Globally Distributed Content Delivery", IEEE Internet Computing, **(2002)** September-October, pp. 50-58.

[9] C. Li, W. Liu and K. Okamura, "A Greedy Ant Colony Forwarding Algorithm for Named Data Networking", Proc. Asia-Pacific Advanced Network, ISSN 2227-302, vol. 34, **(2012)**, pp. 17-26.

[10] B. Krishnamurthy, C. Wills and Y. Zhang, "On the use and performance of content distribution networks", Proc. 1st ACM IGCOMM Internet Measurement Workshop (IMW'01), San Francisco, CA, **(2001)**.

[11] J. Li, S. Verma and K. Ramaswamy, "Centralized scheduler for content delivery network", October 04, 2007, [Online] Available: http://www.google.co.in/patents/WO2007111588A1?cl=en,Publication Number: WO2007111588 A1.

[12] K. Alsabti, S. Ranka and V. Singh, "An efficient k-means clustering algorithm", Electrical Engineering and Computer Science, L.C. Smith College of Engineering and Computer Science at SURFACE, Paper 43. Available: http://surface.syr.edu/eecs/43.

[13] A. A. Gurav and M. J. Nene, "Optimal Path Identification using Ant Colony Optimisation in Wireless Sensor Network", Computer Science & Information Technology, © CS & IT-CSCP 2013, **(2013)**, pp. 223-232.

Authors

Debabrata Sarddar, Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, INDIA. He has done Ph.D. at Jadavpur University. He completed his M. Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He has published around 200 research papers in different journals and conferences. His research interest includes wireless and mobile system, Cloud Computing and WSN.

Sougata Chakraborty is a Senior System Engineer at IBM India Private Limited in Kolkata. He completed M. Tech in Computer Science & Engineering from Jadavpur University in 2011. He had also completed his B. Tech in Information Technology from Murshidabad College of Engineering & Technology under West Bengal University of Technology in 2008. His research interests include Cloud Computing and Mobile Computing.

Priyajit Sen is presently pursuing M. Tech in Computer Science and Engineering at the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India. He has completed his MCA from Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India in 2015. His research interest includes Mobile Computing, Wireless Sensor Network and Cloud Computing.