International Journal of Future Generation Communication and Networking
Vol. 13, No. 2, 2020 pp.980-987

An Efficient Edge Server Deployment Technique In Edge Computing Using Graph Theory

[1] B.Parvathi Devi and [2] Dr.V.Vallinayagi
[1]Reserch Scholar, Manonmaniam Sundaranar University,Tirunelveli.
[2] Vice Principal & Associate Professor, Dept. of Computer Science, Sri Sarada College for Women, Tirunelveli

Abstract

Cloud computing is a paradigm used to store and share the data in the data centers to the users through the Internet. It provides many services to the users like IaaS(Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service) etc , but distributing the centralized data over the world is a complex process, it takes insufficient time. To overcome this, a new paradigm Edge Computing was introduced. Here the data are processed in the edge of the network near the source of the data, so data processing speed is increased. In Edge computing there was no standard architecture followed but some accepted architectures were allowed to process the data. This paper focuses the three layered architecture (Edge nodes, Edge servers, clouds are the main components of this architecture). This paper proposes the placement of new Edge server in the distributed clustered Edge data servers in the middle layer. For placethenew Edge server, here Dijkstra Algorithm is used to find the minimum distance in the architecture and then place the new server. Placing the Edge server in that particular location will improve the performance of the distributed Edge servers, since data travelling distance is reduced and losing the data is also reduced to minimum level. Dijkstra is a famous graph theory algorithm which is mainly used to find the minimum distance between the nodes in graph theory concept. Its performance also analyzed with Floyd-Warshall algorithm. Hope this paper leads to some other techniques to find the minimum distance to place the edge server.

Index Terms:Clustering, Edge Computing, Edge Servers, Dijkstraalgorith,graph based algorithm.

NOMENCLATURE

IaaS -Infrastructure as a Service, PaaS -Platform as a Service, SaaS -Software as a Service, SDN -Software Defined Networking
ESP –Edge Server Placement

1.0 INTRODUCTION

Ten years back the centralized cloud data centers used for processing and sharing the data among many users.[1] But it leads to single point of failure will reflect in the service drop out. So to avoid such circumstance some new paradigms were introduced, one among them is Edge computing.[15] The main benefit of this edge computing is its speed and reliability. Many researches going in this area, because of its computation speed and delivery of data in fraction of seconds.The objective of this paper is to minimize the edge server access delay, and the problem here is, how and where to place the edge servers. In this paper theedge devices are connected to the nearest edge server and request the data which they need and get the replay, they are not directly connected to the cloud data centers.Here the algorithm ESP (Edge Server Placement) do the placement of servers closer to the cloud. This algorithm calculates the shortest path from cloud to the edge using the graph theory concept. Using this algorithm many edge servers are placed according to the users need. So Quality as a Service (QaaS) [2] performance increased, it is the main service of Cloud

ISSN: 2233-7857 IJFGCN

edge computing.

Fig 1. Shows the Edge computing paradigm, which contains the cloud with data centers in the top level and Edge servers are placed in the middle level at the bottom level we have pcs, mobile phones, watches, and laptops etc as edge devices. The Edge devices request the data from the edge servers and they bring the data to them. So the devices not directly connected to the cloud data centers, so time delay will be minimum and the processing speed increased.In the proposed model we have to place the edge servers nearer to the edge devices using the graph theory concept. [3].
The paper is structured as follows, In section 2 some related works are analyzed. In section 3 proposed model and Edge server placement algorithm is illustrated and section 4 of this paper concluded with its future scope.

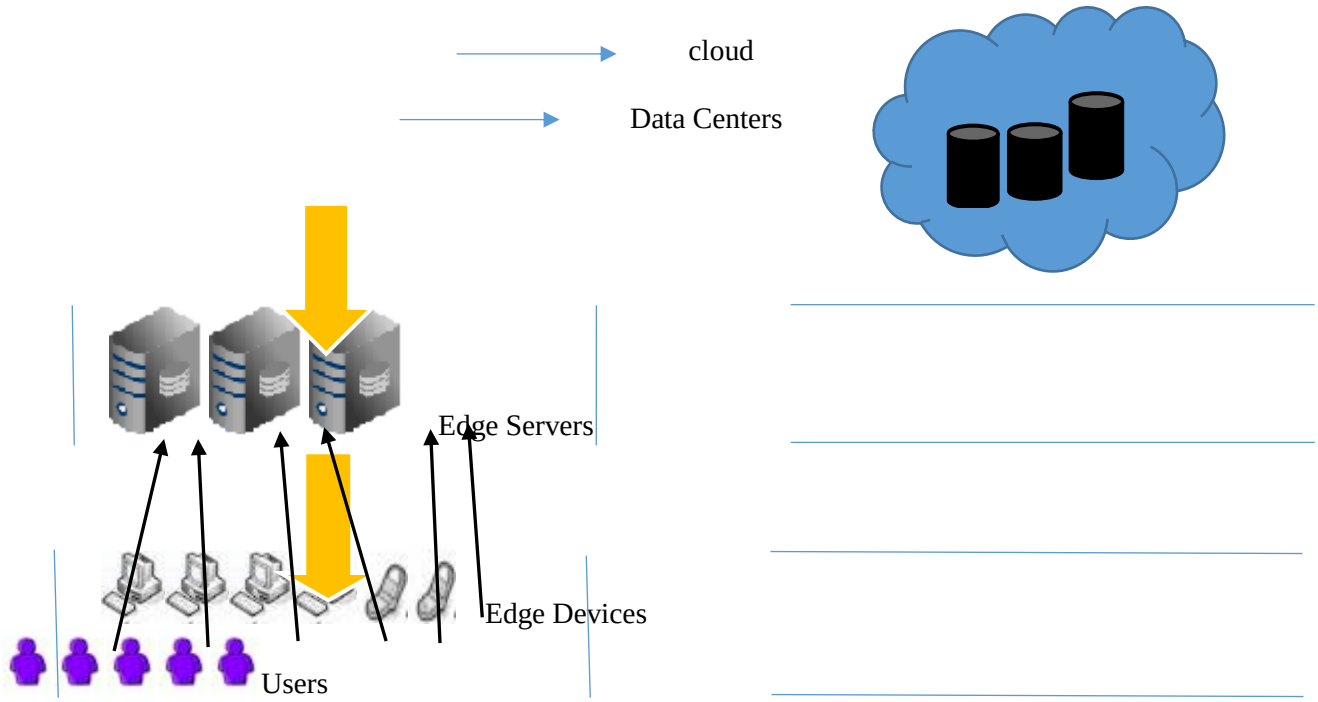


Fig 1. Edge computing paradigm

2. 0 RELATED WORK

Few studies have focused on the edge server placement in Mobile Edge Computing, Cloudlet and Fog computing environments. It is the best work done in the Edge computing paradigm. Currently cloudlet placement in the Mobile Edge computing environment is popular because the work done in offline mode, and here the computers are consider as cloudlet [4] [5]

R.Kannigadevi and G.Murugaboopathy proposed a new clustering model in the cloud data center namely cluster load balancing (CLB) algorithm, which is used to overcome the barriers in the existing random cluster head selection methodologies. The graph based K-means clustering and k-spanning tree clustering algorithms are used to improve the performance, in terms of clustering time, and load distribution is also explained [6]
Shangguang Wang, Yali Zhao et al. proposed an algorithm for edge server placement model for mobile edge computing. The main objective of this algorithm is balancing the workload between edge servers and reduces the edge server access delay. Finally the performance of the proposed algorithmwas analyzed.[7]

ISSN: 2233-7857 IJFGCN

Galluccio [16] done an approach using Prim's algorithm for clustering multi- dimensional data to construct a minimal spanning tree using K-means algorithm
Fan et al. [17] proposed a new strategy, which concentrate the cost and the placement of cloudlet by the CAPABLE strategy to determine the placement of cloudlet so as to optimize the transaction among the cloudlet cost and end-to-end delay.
Zhao et al. [18] go through the cloudlet placement problem for minimized the access delay in SDN (Software Defined Networks) based IOT networks and propose EOPA and RNOPA as the solution to solve this problem.
Heller et al [19] proposed three algorithms silhouette, Gap statistics and PAM to solve the controller placement problems in SDN architecture. Before define the SDN network the optimal number of controllers are placed using these algorithms to reduce the cost of installing SDN controllers.

3. 0 PROPOSED WORK

In the section 3.1 we first introduce the model and section 3.2 we define the algorithm to placement of edge server. Before that some notations are used in the proposed algorithm they are shown in the Table 1

Table 1. Notations

Symbol	Meaning
G(V,E) Connections	Graph with vertices and edges v-servers e-
K	Number of cluster heads
S	Edge server to be placed

3.1 Proposed Model

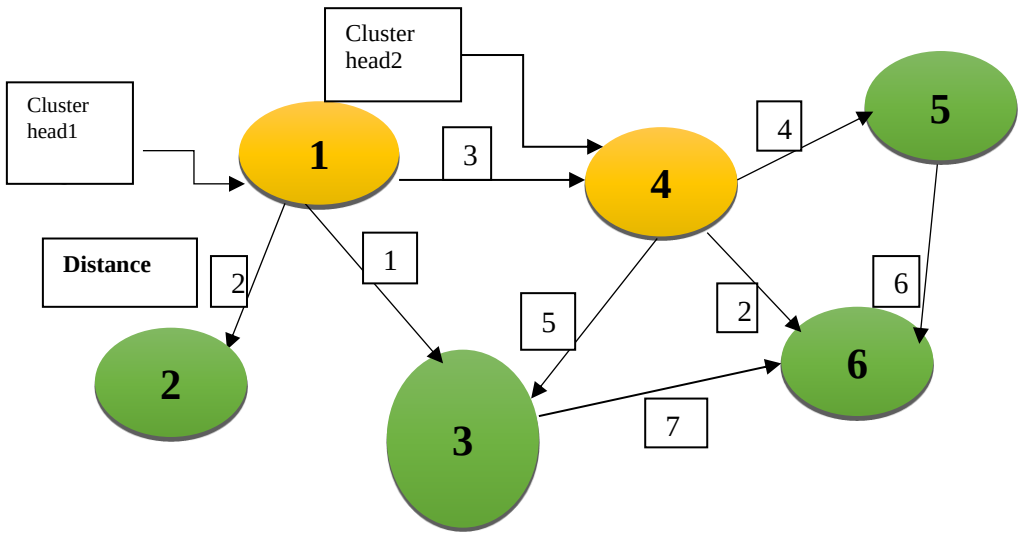


Fig 2. Graph Model of Edge servers

In Fig2. We have a graph representation of clustered edge servers.[8]Consider each node as a edge server.

ISSN: 2233-7857 IJFGCN

It is a weighted graph, the weights are the distance among the edge servers. Here we have 6 edge servers, in that 1 and 4 are cluster heads because they have the maximum degree of 3.

3.2 Dijkstra function

Before going to place the edge server we must know about the mechanism of Dijkstra algorithm. This is the famous algorithm in graph theory concept used to calculate a shortest path from initial node to the destination.[21,22,23] First we assign initial values to the current node and visit the unvisited node and calculate the distance by adding the values. [20] In this way we must calculate all the unvisited nodes distance from the current node position, then compare the calculated values and find the shortest path.

ALGORITHM 1 Dijkstra function

Input : Graph, source(G,S), n

Output : Shortest path

```
1.  for each v in G,
2.  distance[v]= infinity
3.  distance [s]=0
4.  n= no of nodes in the graph
5.  while n != 0
6.  u=node in n with shortest distance[]
7.  remove u from n
8.  for each v of u
9.  temp = distance[u]+distance between (u,v)
10. if temp< distance [v]
11. distance[v]= temp
12. previous[v]=u
13. return previous[]
14. end for
```

In the above function first we initialize all the node's distance as infinity except starting node with 0. Starting node is permanent and it is the active node. Calculate all other neighbor nodes temporary distance by summing its weights with active node. If the calculated distance is smaller, then update the distance and set that node as current node or active node. Mark its distance is permanent. Repeat the steps and find the smallest path.

Now we want to place a edge server in the above fig.2 [13][14] for that we find the shortest distance or nearest path from the cluster head to other server.[9] for example we have a edge server 7 with weight 3 which is to be placed in the path 1-3 means we have two routes 1-3 with distance 1 and 1-4-3 with distance 3+5=8. So the shortest path is 1-3 with distance 1 so place the edge server in that path.First we have to find the distances of all the servers and select a path then place it.

3.3 Proposed Algorithm

In the algorithm1 first we have the input graph G(V, E) (V-Edge server, E-distance) [10] and the k number of cluster head edge servers. We have to check the individual cluster heads and the servers with minimum distance.[11][12] Then place the server in the particular path. In our example we have to place the edge server 7. So we have to check the 2 cluster heads (k). we find the minimum distance edge server 3 in the

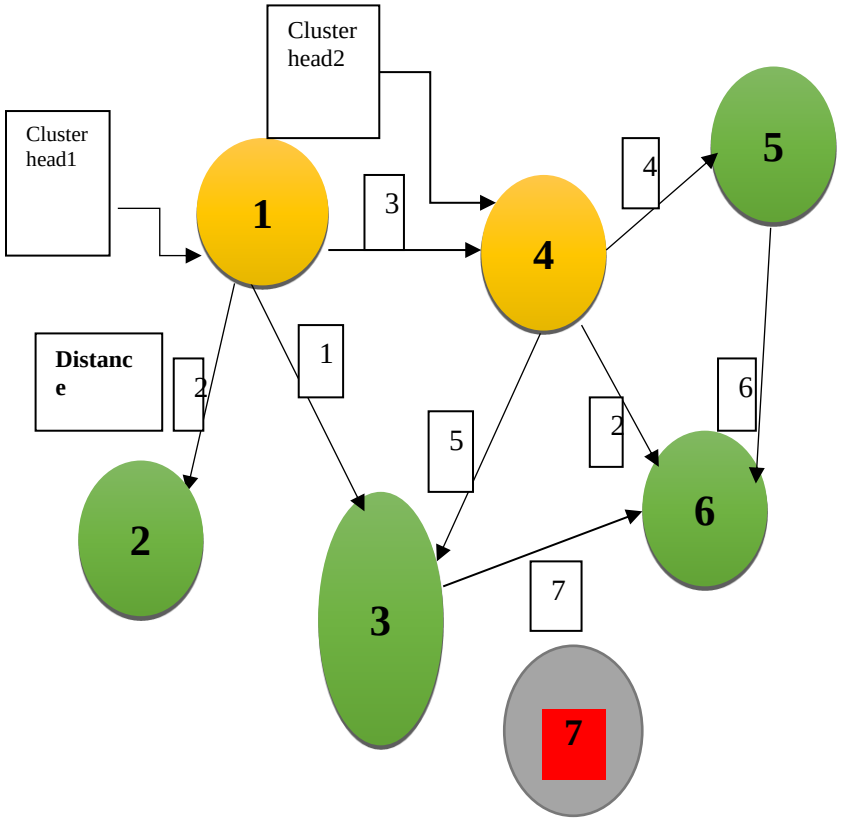
ISSN: 2233-7857 IJFGCN

cluster head 1 with the distance 1 so place 7 in the path 1-3-7. For finding the minimum distance we use Dijkstra algorithm with optimal time complexity.

ALGORITHM 2 Edge Server Placement (ESP) algorithm

Input : G (V, E), K
Output : Edge Server (S) Placement to G
1. G← S
2. for i← 1,K
3. cluster (i) =0
4. call**Dijkstra()**
5. Si ← find nearestserver()
6. if (cluster (i)<=k) then
7. G<- Addserver S
8. cluster (i++)
9. else
10. Si<-find nearestserver()
11. end for

Fig 3. Placement of Edge server using ESP algorithm



ISSN: 2233-7857 IJFGCN

4.COMPARATIVE ANALYSIS OF THE DIJKSTRA ALGORITHM

In Graph theory concept we have many algorithms to find the shortest path, constructing minimum spanning tree etc. In that Prims and KJruskal algorithms re used to find the shortest path between the nodes. But these algorithms differ with time complexity, throughput etc. Here we compared the Floyd-Warshall with Dijkstra algorithm. Both Dijkstra and Floyd-Warshall algorithms find the optimal shortest path by comparing all the possible paths with other nodes, but compared to Dijkstra Floyd-Warshall algorithm works slowly [24]

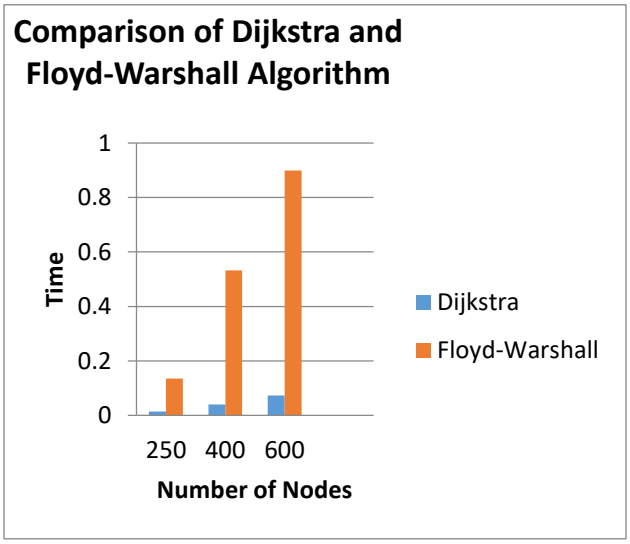
Table 2 Time complexity of Dijkstra and Floyd-Warshall algorithm

Name of the Algorithm	Time complexity
Dijkstra	O(v^2)
Floyd-warshall	O(v^3)

Taking to this constraint we checked the efficiency with different nodes for Dijkstra and Floyd-warshall algorithms. The values are given in the table 3 and its analysis chart also shown below.

Table 3 Comparison of Dijkstra and Floyd –Warshall algorithm

Number of nodes	Dijkstra	Floyd-warshall
250	0.01431	0.1348
400	0.04	0.5321
600	0.07321	0.8991



5. CONCLUSION AND FUTURE SCOPE

This paper presents graph theory solution for placing the Edge server in the clustered distributed edge servers in the edge computing environment. A new algorithm ESP is developed to calculate the minimum distance among the servers and place the server in that particular location. Thus the proposed algorithm improves the service delivery to the users. In Edge computing edge servers plays the main role. So their path must be chosen correctly, otherwise the concept of edge computing is waste. This paper gives one solution to the placement of edge server in the minimum path, and the proposed algorithm uses the Dijkstra algorithm which is comimplemented and In future this algorithm is modified with more secured server placement in the cloud architecture. .Hope this paper leads to some other solution in the research field of Edge computing.

ISSN: 2233-7857 IJFGCN

APPENDIX

APPENDIX A

The Algorithm 1 and 2 are implemented in the MAT LAB and executed with the proper result. It is very easy to learn by all. The modifications also implemented easily in the MATLAB. The intruder detection is the future scope of this paper, so finding the intruder using MAT LAB is very easy and helpful to the beginners.

ACKNOWLEDGEMENT

We thank 3 reviewers for their comments on our first manuscript submitted. It is more helpful for us to correct our errors and we get more knowledge in our research area.

REFERENCES

1. Liu Y, Muppala JK, Veeraraghavan M, Lin D, Hamdi M. *Data Center Networks—Topologies, Architectures and Fault Tolerance Characteristics*.Springer International Publishing; 2013.
2. Khethavath P, Thomas JP, Chan-tin E. Towards an efficient distributed cloud computing architecture. *Peer-To-Peer Network Appl*. 2017;10(5):1152-1168.
3. Garcia Lopez, Pedro; Montresor, Alberto; Epema, Dick; Datta, Anwitaman; Higashino, Teruo; Iamnitchi, Adriana; Barcellos, Marinho; Felber, Pascal; Riviere, Etienne (2015-09-30). "Edge-centric Computing: Vision and Challenges""Edge-centric Computing: Vision and Challenges". ACM SIGC
4. Riaz F, Ali KM. Applications of graph theory in computer science, Proceedings of IEEE 3rd International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN) 2011:142-145
5. Bondy JA, Murty USR. *Graph Theory with Applications*. Elsevier Science Publishing; 1976.
6. Bolhuis M. A comparison between centralized and distributed cloud storage data-center topologies. Proceedings of 19th Twente Student Conference on IT 2013
7. C.C.T. Mark, D. Niyato, T. Chen-Khong, Evolutionary optimal virtual machine placement and demand forecaster for cloud computing, in: 2011 Proceedings of the Advanced Information Networking and Applications, pp. 348-355.
8. Z. Xu, W. Liang, W. Xu, M. Jia, S. Guo, Efficient Algorithms for Capacitated Cloudlet Placements, IEEE Transactions on Parallel and Distributed Systems, 27 (2016) 2866-2880
9. W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal, 3 (2016) 637-64
10. Galluccio L, Olivier Michel PC, Alfred OH. Graph based *K*-means clustering. *Signal Proces*. 2012;92(9):1970-1984
11. Fan, Q.; Ansari, N. Cost aware cloudlet placement for big data processing at the edge. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
12. Zhao, L.; Sun, W.; Shi, Y.; Liu, J. Optimal placement of cloudlets for access delay minimization in sdn based internet of things networks. *IEEE Internet Things J.* **2018**, *5*, 1334–1344.
13. MLAHeller, B.; Sherwood, R.; Mckeown, N. The controller placement problem. *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *42*, 473–478
14. F. B. Zhan and C. E. Noon, “Shortest Path lgorithms: AnEvaluation Using Real Road Networks”, *TransportationScience*. Vol.32, pp.65-73, February 1, 1998
15. Leong Hou U, Hong Jun Zhao, Man Lung Yiu, Yuhong Li, andZhiguoGong,”Towards Online Shortest Path Computation”,IEEE Transactions on Knowledge and Data Engineering,Vol. 26, No. 4, April 2014.

ISSN: 2233-7857 IJFGCN

16. Hua Wu; Marshall, A.; Yu, W., "Path Planning andFollowing Algorithms in an Indoor Navigation Model forVisually Impaired", Internet Monitoring and Protection,2007. ICIMP 2007. Second International Conference on,Vol., No., pp.38, 38, 1-5 July 2007.
17. Hongmei Wang; Ming Hu; Wei Xiao, “A new publictransportation data model and shortest-path algorithms”,Informatics in Control, Automation and Robotics (CAR),2010 2nd International Asia Conference on, Vol.1, No.,pp.456,459, 6-7 March 201
18. Bogdan popa and Dan popescu “Analysis of algorithms for shortest path problem in parallel “ IEEE 2016 17th International Carpathian Control Conference (ICCC) on 29th May. Pg no 613-617.

ISSN: 2233-7857 IJFGCN