

An Efficient Cloud Data Center Allocation to the Source of Requests

Kanniga Devi R., Kalasalingam University, India

Murugaboopathi Gurusamy, Kalasalingam University, India

Vijayakumar P., University College of Engineering, Tindivanam, India

ABSTRACT

A Cloud data center is a network of virtualized resources, namely virtualized servers. They provision on-demand services to the source of requests ranging from virtual machines to virtualized storage and virtualized networks. The cloud data center service requests can come from different sources across the world. It is desirable for enhancing Quality of Service (QoS), which is otherwise known as a service level agreement (SLA), an agreement between cloud service requester and cloud service consumer on QoS, to allocate the cloud data center closest to the source of requests. This article models a Cloud data center network as a graph and proposes an algorithm, modified Breadth First Search where the source of requests assigned to the Cloud data centers based on a cost threshold, which limits the distance between them. Limiting the distance between Cloud data centers and the source of requests leads to faster service provisioning. The proposed algorithm is tested for various graph instances and is compared with modified Voronoi and modified graph-based K-Means algorithms that they assign source of requests to the cloud data centers without limiting the distance between them. The proposed algorithm outperforms two other algorithms in terms of average time taken to allocate the cloud data center to the source of requests, average cost and load distribution.

KEYWORDS

Graph-Model for Cloud Data Center, Modified Breadth First Search, Modified Graph-Based K-Means Algorithm, Modified Voronoi Algorithm, Random Graph Generator

INTRODUCTION

Cloud data centers are the main source of variety of services ranging from computational to network and are delivered as on-demand services to users. The requests for these cloud services can come from different parts of the world (Rawal et al., 2011 and Rawal et al., 2013). The term source of requests/clients denote the users who make requests to various cloud data center services (Shen et al., 2017; Shen et al., 2016). The distance between the cloud data center and the source of requests is a major factor influencing the quality of service in terms of response time and latency. Cloud data center allocation is one of the major issues in cloud computing. An efficient allocation of cloud data

DOI: 10.4018/JOEUC.2020070103

This article, originally published under IGI Global's copyright on July 1, 2020 will proceed with publication as an Open Access article starting on January 21, 2021 in the gold Open Access journal, Journal of Organizational and End User Computing (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

center to the source of requests may improve the quality of services. However, there have been only few approaches that consider the cloud data center allocation to the source of requests.

In recent literature, Joseph Doyle et al. (2013) has proposed the source of requests assignment to the closest cloud data center to reduce the carbon emission, but they modeled cloud data center as a complete graph, which is unrealistic. They modelled both the networking and computational components of the infrastructure as a graph and proposed a system which utilizes Voronoi partitions to determine how source requests to be routed to appropriate data center based on the relative priorities of the cloud operator for latency purposes. This allows routing of the traffic to the data center that is closest in terms of geographical distance, costs the least in terms of power, and emits the smallest amount of carbon for a given request to lower carbon emissions and operational cost. This work examined the electricity cost, carbon emissions, and average service request time for a variety of scenarios.

(Judith Bar-Ilan et al. (1992), Randeep Bhatia et al. (1998), Reza Zanjirani Farahani et al. (2010), Irina Harris et al. (2014) provided solutions for facility location problems. They have considered distributing the clients to centers as balanced as possible, but they have overlooked the distance between clients and centers, which is also essential for faster service provisioning, hence there arise a need to develop an efficient method which allocates closest cloud data centers to the source of requests and keeps the load of the cloud data center as balanced as possible.

This paper models cloud data center as a graph and proposes an algorithm - modified Breadth First Search (MBFS) to efficiently allocate cloud data centers to the source of requests based on a cost threshold. Here the term cost refers to the distance between the cloud data center and the source of request. The aim is to allocate each source of request to a cloud data center in a faster manner based on cost threshold. The cost threshold is calculated as the average path length between cloud data centers and the source of requests of modified Voronoi approach. This may lead to faster service provisioning of the cloud data centers to the source of requests. The performance of proposed algorithm is compared with that of modified Voronoi and modified graph-based K-Means algorithms for various graph instances.

The following contributions are made in this paper:

1. Modified Breadth First Search algorithm is proposed;
2. Modified Voronoi algorithm is proposed;
3. Modified graph-based K-Means algorithm is proposed;
4. A random graph generator is constructed;
5. Comparison between approaches 1, 2 and 3 is done in terms of average time taken for allocation, average cost of cloud data centers and average load of cloud data centers.

RELATED WORK

In recent literature, Joseph Doyle et al. (2013) has proposed the source of requests assignment to the closest cloud data center to reduce the carbon emission but they modeled cloud data center as a complete graph which is unrealistic. They modelled both the networking and computational components of the infrastructure as a graph and proposed a system which utilizes Voronoi partitions to determine how source requests to be routed to appropriate data center based on the relative priorities of the cloud operator for latency purposes. This allows routing of the traffic to the data center that is closest in terms of geographical distance, costs the least in terms of power, and emits the smallest amount of carbon for a given request to lower carbon emissions and operational cost. This work examined the electricity cost, carbon emissions, and average service request time for a variety of scenarios. Judith Bar-Ilan et al. (1992) dealt with the issue of allocating and utilizing centers in a distributed network. This work proposed some approximation algorithms for selecting centers and distributing the users among them. They considered balanced versions of allocation where the

assignments of clients to centers are as balanced as possible. They dealt with balanced and weighted version of K-center problem and ρ -dominating set problem. Randeep Bhatia et al. (1998) proposed that generally facility location problems have been studied in networks with static edge lengths. The network could be used to model any general real-life facilities ranging from hospitals to information centers. Due to traffic congestion the transit time on links changes over time. This work estimated how the edge lengths change with time, and are choosing a set of locations as centers, such that at every time instant each vertex has a center close to it. It also provided approximation algorithms for the K-center problem under this model. Though the above two literatures seem very, their concepts still find applications in many areas of computer science and engineering. Reza Zanjirani Farahani et al. (2010) provided a review on facility location problems in three categories including bi-objective, multi-objective and multi-attribute problems and their solution methods. This work proposed that multi-criteria facility location problems achieve closer solutions to reality as it considers more criteria. This work mainly reviewed the facility location problems applied to real world problems. Based on the review, it suggested location-reliability problems as facilities are prone to natural/man-made disasters. It also suggested that usage of random parameters would be more realistic. It suggested the applications of multi-criteria location decision with routing, inventory and transportation, etc. Irina Harris et al. (2014) proposed a multi-objective optimization approach to the capacitated facility location–allocation problem (CFLP). At the allocation level for large instances, financial costs and CO₂ emissions were considered simultaneously where Lagrangian Relaxation models for dealing with costs and CO₂ emissions at the location level. Thus, this method assessed the robustness of each location solution with respect to two objectives for customer allocation. Anupama Potluri et al. (2012) proposed that the Minimum Capacitated Dominating Set (CAPMDS) problem and the capacity can be uniform or variable across all the nodes. Being a generalization of the Minimum Dominating Set problem, this problem is NP-hard. This work presented a heuristic and a couple of its variants for solving the CAPMDS problem. These heuristics work for both uniform and variable capacity graphs. Anupama Potluri (2012) conducted an empirical study of the Minimum Independent Dominating Set (MIDS) problem to understand the practical significance of the Polynomial-Time Approximation Schemes (PTAS) algorithm. This work compared the solutions of PTAS with those of various heuristics for MIDS. For the graph instances, PTAS solutions were as good as those returned by the best heuristic. In many instances, the PTAS returned solutions worse than those of the best heuristic. The time taken by the PTAS was high even for small graphs. It is impractical for large and high-density graphs as it uses an exact algorithm to compute the local optimal dominating set. This is the first work to propose metaheuristic algorithms for the CAPMDS problem which are better than heuristics and approximation algorithms. Server consolidation during off-peak hours leads to great energy efficiency in data center networks, which requires switching off some of the physical machines. Some works have been proposed so far without considering network traffic patterns between VMs, which can lead to inefficiency. Geetha Sowjanya Akula et al. (2014) proposed an algorithm that considers migration of Virtual Machines (VMs) as an ensemble based on traffic pattern. Laurent Galluccio et al. (2012) proposed an approach for clustering multi-dimensional data used Prim's algorithm to construct a minimal spanning tree (MST), and thresholding the sequence of edge lengths in order to determine both the number of clusters and the locations of the cluster centroids.

These literature surveys directed us to a different dimension on limiting the distance from the cloud data centers to the source of requests to speed up the service provisioning. So, there arise a need to develop an efficient method to allocate cloud data centers to the source of requests.

PROPOSED METHOD

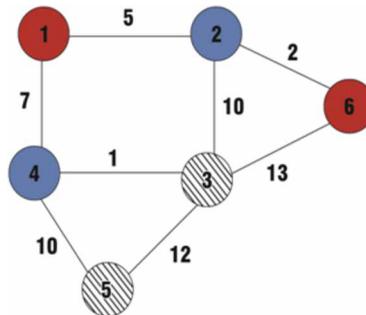
The cloud data center infrastructure is modeled as a graph and a modified Breadth First Search is used for efficient allocation of cloud data centers to the source of requests.

Modelling Cloud Data Center Infrastructure as a Graph

The cloud data center infrastructure is modeled as an undirected weighted graph where some of the nodes are designated as data centers and remaining nodes represent the source of requests. The edge weight denotes the distance between pair of nodes.

In Figure 1, nodes 1 and 6 denote cloud data centers and nodes 2, 3, 4, 5 denote the source of requests.

Figure 1. Cloud data center infrastructure graph



Modified Breadth-First Search

This section depicts how the modified Breadth First Search is used to efficiently allocate cloud data centers to the source of requests.

Breadth-first search (Cormen et al., 2009) is one of the algorithms for searching a graph. Given a graph $G=(V,E)$ and a source vertex s , this search examines the edges of G to find every vertex that is reachable from s and computes the distance between s and each reachable vertex. Normally, BFS is applied only on unweighted graphs.

The Breadth First Search (BFS) has been modified to reflect the problem of cloud data center allocation to the source of requests. Instead of just considering shortest path from a source vertex, the proposed algorithm performs BFS on selected vertices which are declared as cloud data centers and covers nodes lying within the cost threshold, thereby limiting the walk from cloud data centers to source of requests. The modified Breadth First Search algorithm (MBFS) is mentioned below.

Cloud Data Centers-Source Requests Coverage Algorithm Using MBFS

The cloud data center allocation problem is modeled using an undirected weighted graph as depicted in Figure 1, where the weights represent distance between nodes. The set of cloud data centers are declared in advance and the cost threshold is also set based on the average path length between cloud data centers and the source of requests of Voronoi approach. The MBFS is applied on each data center to find the source of requests covered within the cost threshold by each cloud data center.

The MBFS assumes that the input graph $G=(V,E)$ is an undirected weighted graph and represented using adjacency matrix. It considers additional attributes to each vertex in the graph. It holds color as an attribute for each vertex. The attribute *dist* stores the distance from the data center d to each vertex u . The *vertex_covered* stores the list of vertices covered by the cloud data center vertices. The algorithm uses a queue to store the set of gray-colored vertices.

For example, in Figure 1, vertices 1 and 6 are declared as cloud data centers. When the algorithm is applied on these vertices, it returns the vertices covered within the cost threshold by both the cloud data centers as follows:

The vertices covered by vertex 1 are:

2, 4

The vertices covered by vertex 6 are:

none

Cloud Data Centers-Source Requests Coverage Algorithm

Algorithm: MBFS

Input: $G=(V,E,w), w: E \rightarrow R, DC \subseteq V$, where $1 \leq |DC| \leq k$, $\rho = \text{cost_threshold}$

Output: $\text{vertex_covered} \forall DC$

1. **for** each $d \in DC$ **do**
2. $\text{vertex_covered} = \text{modified_BFS}(G, d, \rho)$
3. Print vertex_covered

vertex_covered modified_BFS(G,d, ρ)

1. $\text{vertex_covered} = \{\}$
2. $Q \leftarrow \{\}$
3. **for** each u in $V - \{d\}$
4. $\text{color}[u] \leftarrow \text{WHITE}$
5. $\text{dist}[u] \leftarrow \text{infinity}$
6. $\text{color}[d] \leftarrow \text{GRAY}$
7. $\text{dist}[d] \leftarrow 0$
8. $\text{ENQUEUE}(Q, d)$
9. **while** Q is non-empty
10. $u \leftarrow \text{Front}[Q]$
11. **for** each v adjacent to u
12. **if** $\text{color}[v] \leftarrow \text{WHITE}$ and $(\text{dist}[v] = \text{dist}[u] + \text{dist}[u, v]) \leq \rho$
13. **then** $\text{color}[v] \leftarrow \text{GRAY}$
14. $\text{ENQ}(Q, v)$
15. $\text{DEQ}(Q)$
16. $\text{color}[u] \leftarrow \text{BLACK}$
17. $\text{vertex_covered} = \text{vertex_covered} \cup \{u\}$
18. **return** $\text{vertex_covered} - \{d\}$

Each cloud data center performs walk based on cost threshold in order to cover the source of requests. Still, some sources of requests are not assigned to any Cloud data center at all. To deal with these leftover source of requests, two approaches has been followed. The first approach assigns leftover source of requests to some cloud data centers randomly. The second approach assigns leftover source of requests to minimum distant data center or to a neighbour which has minimum distance to cloud data center.

Complexity of the Algorithm

The complexity of MBFS implemented using an adjacency matrix will be $O(|V|^2)$ where V is the number of vertices. This is mainly because every time it needs to find what are the edges adjacent to

a given vertex u and traverse the whole array $\text{AdjMatrix}[u]$, which is of course of length $|V|$, hence the total time would become $O(|V| * |V|)$ which is $O(|V|^2)$.

VORONOI APPROACH TO CLOUD DATA CENTER ALLOCATION

Voronoi approach on graphs can be used to find shortest paths between cloud data centers and the source of requests to help in efficient allocation.

Voronoi Partition

Voronoi partitions (Doyle et al., 2013) are the decomposition of set of points into subsets. These subsets are centered around points known as sites. Each point in the set is added to a subset consisting of a site and all other points associated with this site. An abstract notion of distance between a point and the sites is used to determine, which subset a point is associated with. A point is assigned to a subset if the distance to a site is less than or equal to the distance to the other sites. (Durham et al., 2012) proposed partitioning and coverage control algorithm based on Voronoi approach for a network of robots, where each partitioned region will be covered by an individual robot. (Doyle et al., 2013) proposed the Stratus system, which utilizes Voronoi partitions to determine, how the source requests to be routed to appropriate cloud data center based on the priorities of the cloud operator. The Stratus system has a complete graph structure, where there is a single edge from the cloud data centers to the source of requests, whereas in the proposed scheme, there is a path from the cloud data centers to the source of requests.

Voronoi Partitions of the Cloud

In the graph, Voronoi algorithm based on shortest paths is applied on the cloud data centers to find the minimum distance to the source of requests. Then the source of requests are allocated to the cloud data centres to which the distance is minimum.

Cloud Data Centers Assignment to Source Requests Using Voronoi Approach

It finds the minimum distant paths on a weighted graph $G=(V,E)$. Therefore, $w(u,v) \geq 0$ for each edge $(u,v) \in E$. The subset S of V forms the set of data centers DC. The algorithm selects the vertex $u \in V-S$ with the minimum shortest-path, adds u to S , and relaxes all edges leaving u . It uses a queue Q of vertices, storing $dist$ values. Finally, the distance from the cloud data centers to all other vertices are compared and the source request nodes are allocated to the data centres to which the distance is minimum.

Cloud Data Centers Assignment to Source Requests

Algorithm: modified_Voronoi

Input: $G=(V, E, w)$, $w: E \rightarrow \mathbb{R}$, $DC \subseteq V$, where $1 \leq |DC| \leq k$,

Output: $DC \rightarrow V$

1. **for** each $d \in DC$ do
2. modified_Voronoi(G, DC, d, w)
3. **for** each $v \in V$ && ($v \notin DC$)
4. choose $d \in DC$ where $\min_{u \in DC} \{dist(u, v)\}$
5. $d \rightarrow v$

modified_Voronoi(G, DC, d, w)

1. init_source(G, d)
2. Src $\leftarrow \emptyset$
3. $Q \leftarrow V$

```
4. while  $Q \neq \emptyset$ 
5.      $u \leftarrow \text{Min}(Q)$ 
6.      $\text{Src} \leftarrow \text{Src} \cup \{u\}$ 
7.     for each  $v$  adjacent to  $u$ 
8.         do relax_Voronoi( $u, v, w, DC$ )

init_source( $G, d$ )
1. for each  $v \in V$ 
2.      $\text{dist}[v] \leftarrow \text{INFINITY}$ 
3.      $\text{dist}[d] \leftarrow 0$ 
relax_Voronoi( $u, v, w, DC$ )
1. if ( $\text{dist}[v] > \text{dist}[u] + w(u, v)$ ) && ( $v \notin DC$ )
2.      $\text{dist}[v] \leftarrow \text{dist}[u] + w(u, v)$ 
```

Complexity of the Algorithm

The complexity of modified Voronoi implemented using an adjacency matrix is as $O((|V|+|E|) \log |V|)$ in case of priority queue-based implementation.

GRAPH-BASED K-MEANS APPROACH

A Graph-based K-Means (Galluccio et al.,2012) approach has been developed to allocate cloud data centers to the source of requests. K-Means algorithm initially takes K as an input and then partitions the given set of points into K groups.

Procedure of K-Means Algorithm

As clustering of clients to cloud data centers is performed, a modified graph-based K-means clustering algorithm has been developed. Each node in the graph is designated either as a source of request or a cloud data center and weight of an edge as the Euclidean distance between the two nodes. Initially, K nodes are picked and are declared as cloud data centers. Floyd-Warshall algorithm (Bondy et al., 1976) is used to find the distance from each node to all the other nodes in the graph.

The procedure of modified graph-based K-Means clustering algorithm is as follows:

1. From the data set, select K number of points randomly and set them as initial cluster heads;
2. Find the distance from each point in the data set to all the cluster heads;
3. For each point in the data set, assign it to a cluster whichever is nearer to it.

Modified Graph-Based K-Means Approach to Cloud Data Center Allocation

Algorithm: Modified graph-based K-Means

```
Input:  $G=(V, E, w), w: E \rightarrow R, DC \subseteq V, \text{ where } 1 \leq |DC| \leq k,$   
Output:  $DC \rightarrow V$   
1. call FloydWarshall( $G$ )  
2. for  $i \leftarrow 1, k$  do  
3.      $DC[i] \leftarrow \text{ClusterHead}[i]$   
4. end for  
5. for all  $v \in V \wedge v \notin DC$  do  
6.      $C \leftarrow \text{FindNearestCluster}()$   
7.      $C \leftarrow v$   
8. end for
```

Time Complexity

The complexity of graph-based K-means algorithm implemented using an adjacency matrix is as $O(n^3)$ because of Floyd Warshall's algorithm.

EXPERIMENTATION

Implementation of the proposed MBFS algorithm, modified Voronoi algorithm and modified graph-based K-Means has been done to compare their performance in terms of average time taken for allocation, average cost of cloud data centers, and average load of cloud data centers.

Random Graph Generator

A graph generator generating 20 random weighted undirected graph instances for varying number of vertices from 100 to 1000 has been developed. As the graph is dense, an adjacency matrix of the graph is considered. The edge weights are set between 0 and 30 for all the graph instances.

Algorithm: Random graph generator

Input: Number of nodes n .

Output: A graph $G=(V,E)$.

1. $V \leftarrow \{ 1, 2, \dots, n \};$
2. $E \leftarrow \emptyset;$
3. for each $\{i, j\} \subseteq E$, where $i \neq j$ do
4. $w\{i, j\} \leftarrow \text{random}(0, 30);$
5. $E \leftarrow E \cup \{ \{i, j\} \};$
6. return $G=(V, E);$

The cloud data centers are selected from the randomly generated graph instance. The number of cloud data centers is kept same for various graph instances since the cloud data centers are usually set up only once and only the source of requests varies over time. Hence, size of the cloud data centers remains the same even when the size of the graph instance increases.

Cost Threshold Computation

Cost threshold is computed from the average path length between cloud data centers and the source of requests of Voronoi approach. This cost threshold is used to perform walk from the cloud data centers to the source of requests in MBFS.

Let:

m be the number of cloud data centers

n be the number of source of requests

CDC_j be the j^{th} cloud data center, where $1 \leq j \leq m$

$Path\ length_i$ be the path length of i^{th} source of request, where $1 \leq i \leq n$

$$\text{Average path length of } CDC_j = \sum \text{pathlength}_i / n \quad (1)$$

$$\text{Average path length across CDC} = \sum CDC_j / m \quad (2)$$

Hence, cost threshold of MBFS is equal to average path length across cloud data centers of Voronoi approach. In MBFS, cloud data centers perform walk using this cost threshold leaving some

source of requests not covered by any cloud data center at all. The following section illustrates the assignment of such leftover nodes to cloud data centers.

Cloud Data Center Allocation Methods to the Source of Requests

There are two ways to assign cloud data centers to the source of requests in MBFS:

- Usage of average path length as cost threshold by the cloud data centers for covering source of requests and randomly assigning leftover source of requests to cloud data centers;
- Usage of average path length as cost threshold by the cloud data centers for covering source of requests and assigning leftover source of requests to minimum distant cloud data center or to its minimum distant neighbour.

Performance Metrics

In MBFS, cloud data centers perform walk using this cost threshold leaving some source of requests not covered by any cloud data center at all. The following section illustrates the assignment of such leftover nodes to cloud data centers.

Following metrics are used for comparing MBFS approaches with Voronoi approach:

Average Time taken

Let T be the average time taken by the algorithm measured in seconds (s):

Average cost

Let:

m be the number of cloud data centers

n be the number of source of requests

CDC_j be the j^{th} Cloud Data Center, where $1 \leq j \leq m$

$Cost_i$ be the cost of i^{th} source of request, where $1 \leq i \leq n$

$$\text{Average cost of } CDC_j = \sum cost_i/n \quad (3)$$

$$\text{Average cost across } CDC = \sum CDC_j/m \quad \text{Average cost across } CDC = \sum_{j=1}^m CDC_j/m \quad (4)$$

Average Load

Let L be the average number of source of requests assigned to cloud data centers.

RESULTS AND DISCUSSION

This section presents comparison results of modified Breadth First Search, modified Voronoi and modified graph-based K-Means approaches.

Figure 2 shows the average time taken by all the algorithms for various graph instances. The MBFS approach takes less time compared to that of Voronoi and graph-based K-Means approaches. The modified Voronoi and graph-based K-Means approaches are optimal algorithms taking more time as they always find only the minimum distant cloud data center.

Figure 2. MBFS vs. Other algorithms in terms of average time taken

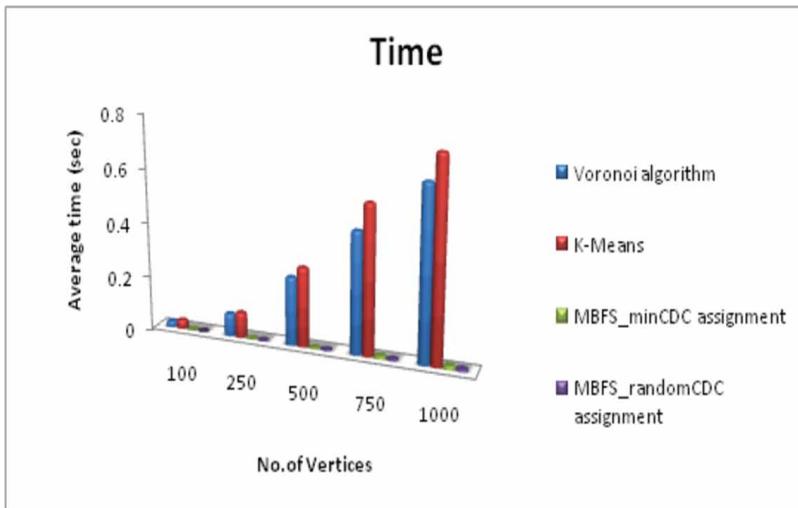
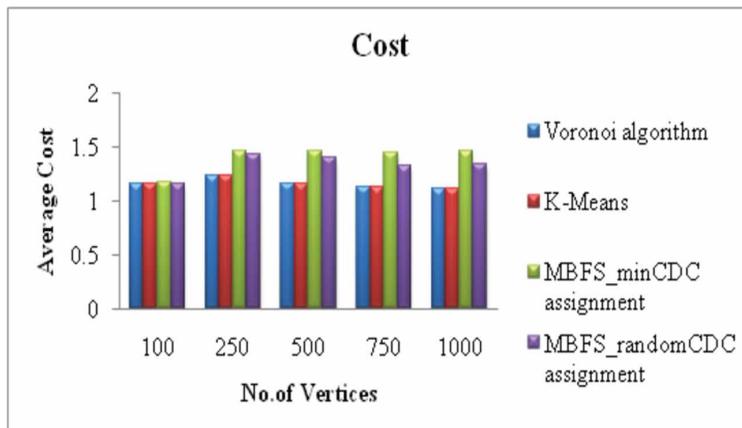


Figure 3 shows that the average cost of cloud data centers of all the algorithms. The average cost of MBFS is more than that of Voronoi and K-Means approaches as MBFS assigns leftover nodes either randomly or to a minimum distant data center in order to service all the source of requests which increases the average cost, whereas Voronoi and K-Means approaches always assign only to

Figure 3. MBFS vs. Other algorithms in terms of average cost



minimum distant data center. Moreover, the cost comes down with larger number of vertices is due to the connectivity (degree of nodes) has increased. Primarily, this would mean that these are dense graphs and so minimum cost of paths will reduce. The difference in average cost between MBFS and other approaches has not shown any significant difference. Hence, MBFS solution can be used for cloud data center allocation to source of requests instead of the time-consuming Voronoi and K-Means approaches.

Table 1 shows that the average load distribution of 100 source of requests to cloud data centers of all the algorithms. The average load of cloud data centers is uniform, as all the source of requests are assigned to some cloud data centers. However, the average load distribution is

Table 1. MBFS vs. Other approaches in terms of average load distribution

No. of DCs	DC ID	Voronoi Algorithm	K-Means	MBFS_minCDC Assignment	MBFS_randomCDC Assignment
25	3	8	5	3	3
	5	4	3	3	3
	15	8	1	3	3
	16	9	7	3	3
	19	6	2	3	3
	25	5	6	3	3
	27	4	3	3	3
	28	0	4	3	3
	29	2	5	3	3
	32	0	4	3	3
	34	1	2	3	3
	39	0	2	3	3
	43	2	2	3	3
	45	2	1	3	3
	46	3	3	3	3
	48	2	2	3	3
	51	1	1	3	3
	52	0	1	3	3
	53	4	4	3	3
	67	4	4	3	3
71	2	3	3	3	
74	1	0	3	3	
90	2	1	3	3	
99	4	6	3	3	
100	1	3	3	3	

uniform in case of MBFS approaches, whereas in Voronoi and K-Means approaches, some cloud data centers are overloaded and some are underloaded as they always assign only to a minimum distant cloud data center.

From the results it is clear that MBFS assigns cloud data center to the source of requests in a faster manner, which leads to faster service provisioning.

CONCLUSION AND FUTURE WORK

This paper proposed an algorithm-modified Breadth First Search for efficiently allocating cloud data centers to source of requests by limiting the distance between them and keep the load of the cloud data center as balanced as possible. The proposed approaches are compared with modified Voronoi and K-Means approach of cloud data center allocation. The proposed algorithm outperforms the modified Voronoi and modified K-Means in terms of average time taken, average cost and load distribution.

In future, there is a plan of augmenting Capacitated Dominating Sets with cloud data center allocation which will further fine tune allocation problems by enabling load balancing of cloud data centers. In the Capacitated Dominating Set problem, each vertex has been assigned a limit on the number of vertices it can dominate. Hence, every cloud data center augmented with Capacitated Dominating Sets will serve uniform source of requests.

REFERENCES

- Akula, G. S., & Potluri, A. (2014). Heuristics for Migration with Consolidation of Ensembles of Virtual Machines. *Proceedings of the 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*. IEEE. doi:10.1109/COMSNETS.2014.6734927
- Bar-Ilan, J., Kortsarz, G., & Peleg, D. (1992). How to allocate network centers. *Journal of Algorithms*, 15(3), 385–415. doi:10.1006/jagm.1993.1047
- Bhatia, R., Guha, S., Khuller, S., & Sussmann, Y. J. (1998). Facility Location with Dynamic Distance Functions. *Journal of Combinatorial Optimization*, 2(3), 199–217. doi:10.1023/A:1009796525600
- Bondy, J.A. & Murty. U.S.R. (1976). *Graph theory with applications*. Elsevier.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). MIT press.
- Doyle, J., Shorten, R., & O'Mahony, D. (2013). Stratus: Load Balancing the Cloud for Carbon Emissions Control. *IEEE Transactions on Cloud Computing*, 1(1), 116–128. doi:10.1109/TCC.2013.4
- Durham, J. W., Carli, R., Frasca, P., & Bullo, F. (2012). Discrete Partitioning and Coverage Control for Gossiping Robots. *IEEE Transactions on Robotics*, 28(2), 364–378. doi:10.1109/TRO.2011.2170753
- Farahani, R. Z., SteadieSeifi, M., & Asgari, N. (2010). Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*, 34(7), 1689–1709. doi:10.1016/j.apm.2009.10.005
- Galluccio, L., Michel, O., Comon, P., & Hero, A. O. III. (2012). Graph based K-means clustering. *Signal Processing*, 92(9), 1970–1984. doi:10.1016/j.sigpro.2011.12.009
- Harris, I., Mumford, C. L., & Naim, M. M. (2014). A Hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. *Transportation Research Part E, Logistics and Transportation Review*, 66, 1–22. doi:10.1016/j.tre.2014.01.010
- Potluri, A. (2012). A Study of Algorithms for Minimum Dominating Set and its Generalizations [Doctoral dissertation]. SCIS, University of Hyderabad, Hyderabad, India.
- Potluri, A., & Singh, A. (2012). A Greedy Heuristic and its Variants for Minimum Capacitated Dominating Set. *Proceedings of the International Conference on Contemporary Computing IC3 2012* (pp. 28-39). Springer. doi:10.1007/978-3-642-32129-0_9
- Rawal, B. S., Karne, R.K., & Wijesinha, A.L. (2011). Splitting HTTP requests on two servers. *Proceedings of the Third International Conference on Communication Systems and Networks (COMSNETS)* (pp. 1-8). IEEE.
- Rawal, B. S., Berman, L., & Ramcharan, H. (2013). Multi-client/Multi-server split architecture. *Proceedings of the International Conference on Information Networking (ICOIN)* (pp. 696-701). IEEE. doi:10.1109/ICOIN.2013.6496712
- Shen, J., Chang, S., Shen, J., Liu, Q., & Sun, X. (2016). A lightweight multi-layer authentication protocol for wireless body area networks. *Future Generation Computer Systems*. doi:10.1016/j.future.2016.11.033
- Shen, J., Liu, D., Shen, J., Liu, Q., & Sun, X. (2017). A secure cloud-assisted urban data sharing framework for ubiquitous-cities. *Pervasive and Mobile Computing*, 41, 219–230. doi:10.1016/j.pmcj.2017.03.013
- Shen, J., Shen, J., Chen, X., Huang, X., & Susilo, W. (2017). An Efficient Public Auditing Protocol with Novel Dynamic Structure for Cloud Data. *IEEE Transactions on Information Forensics and Security*, 12(10), 2402–2415. doi:10.1109/TIFS.2017.2705620
- Shen, J., Zhou, T., He, D., Zhang, Y., Sun, X., & Xiang, Y. (2017). Block Design-based Key Agreement for Group Data Sharing in Cloud Computing. *IEEE Transactions on Dependable and Secure Computing*. doi:10.1109/TDSC.2017.2725953

R. Kanniga Devi received her Bachelor of Engineering Degree from Madurai Kamaraj University, Madurai and Master of Engineering from Anna University, Chennai. She is currently doing her doctoral program from Kalasalingam University, Srivilliputhur. She is a member of CSI, IAEng and ISTE. She has 14 years of teaching experience. She has published 12 research papers in International Journals and Conferences. Her research interest include Cloud computing and Graph theory applications in Cloud Computing. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Kalasalingam University, Tamil Nadu, and India.

G. Murugaboopathi received the Undergraduate Degree in Computer Science and Engineering from Madurai Kamaraj University, Post Graduate degree in Digital Communication and Network from Madurai Kamaraj University and Ph.D in Computer Science and Engineering at Bharath University, Chennai. He has more than 45 publications in National, International Conference and International Journal proceedings. He has more than 14 years of teaching experience. His areas of interest include Wireless Sensor Networks, Mobile Communication, Mobile Computing, Mobile Adhoc Networks, Cloud Computing, Computer Networks, Network Security, Network and Data Security, Cryptography and Network security and etc., He is currently working as an Associate Professor in the Department of Computer Science and Engineering at Kalasalingam University, Tamil Nadu, and India.

P. VijayaKumar completed his Ph.D in Computer Science and Engineering in Anna University Chennai in the year 2013. He completed Master of Engineering in the field of Computer Science and Engineering in Karunya Institute of Technology in the year 2005. He completed his Bachelor of Engineering under Madurai Kamarajar University in the year 2002. He is presently working as Dean at University College of Engineering, Tindivanam. He is guiding for many Ph.D scholars in the field of network and cloud security. He has published various quality papers in the reputed journals like IEEE Transactions, Elsevier, Springer, IET, Taylor & Francis, Wiley etc. His main thrust research areas are Key management in Network Security and Multicasting in Computer Networks.