Load Balancing Framework In Geo-Distributed E-Learning Fog Computing Platform

Ogbonna Chima Otumdi[1]
Department of Computer Engineering
University of Uyo, Akwa Ibom State Nigeria
otumdii@yahoo.com

Ozuomba Simeon[2]
Department of Computer Engineering
University of Uyo, Akwa Ibom State Nigeria
simeonoz@yahoo.com

Philip Asuquo[3]
Department of Computer Engineering
University of Uyo, Akwa Ibom State Nigeria
philipasuquo@uniuyo.edu.ng

***Abstract*— In this paper, load balancing framework in geo-distributed e-learning fog computing platform is presented. The framework is designed to enhance the management of user connections in a geo-distributed fog computing environment, to avoid overloading of fog nodes that may lead to system failure due to computational overhead or underloading of fog nodes that may lead to underutilization of available compute resources. The framework enables a distributed setup thereby allowing for segmentation of users into smaller groups to prevent over subscription on a particular system. Furthermore, the proposed framework consists of three layers; the first layer is the cloud layer that contains information on the fog nodes present in the system as well as registration details of prospective users of the system. The second layer also middle layer is the fog layer that comprises of all the fog nodes in the system and each of the fog node contains the application user interface and also has its set of modules required for operation. Lastly, the third layer is the users' domain. The users are connected based on a systematic recommendation process so as to connect to the most suitable node within the location of the user. The flow diagram and algorithm of the proposed model are presented. Finally, the load balancing framework will improve the performance of the e-learning environment by having a more evenly distributed connection.**

Keywords: Load Balancing Framework, Fog Node, Cloud Computing, E-Learning Platform, geo-distributed architecture

1. INTRODUCTION

E-learning systems are comprised of sub-systems dedicated to handle specific functions in an e-learning ecosystem. Before recent times, e-learning was limited to asynchronous data transfer where learners were limited to text communication via chat systems and bi-directional file transfer of e-learning resource materials. Moreover, at that early stage, multimedia communications was at infancy stage, also limited to asynchronous image transfer, audio communication and video transfer. However, advancement in technology and the global COVID-19 pandemic brought about a boost in synchronous time-sensitive data transfer such as real-time video conferencing amongst others [1][2]. This has moved e-learning systems from offline electronic systems to online real-time interactive systems where learners and the teachers alike interact live.

To meet these current needs in e-learning systems, Cloud Computing (CC) has been adopted by many users in the academia and industry [3][4]. However, coupled with the high financial cost of subscribing to CC, especially in third world countries, its services and processes are performed on centralized shared servers at remote locations over the internet. This necessitates fresh challenges in its application to e-learning systems.

Fog computing (FC), which is an extension of cloud computing, was birthed to bridge the gap between the cloud and the end-user [5]. Specifically, fog computing was designed to negate the drawbacks associated with cloud computing such as issues of high latency in executing time constrained task, high bandwidth consumption, compute resources unavailability due to computational overhead as well as data privacy and ownership matters amongst others. FC which is an emerging technology can be described as a distributed architecture with communication, computing and storage capability at the edge of the network [4]. Basically, it exists at the layer between the cloud and the end user device, while providing cloud-like services. Due to its distributed architecture and proximity to the end user, tasks are processed faster with low latency and response time thereby making it ideal for time sensitive applications [4][5][6]. In view of this advantages, researchers have adopted this FC to implement e-learning systems [7].

However, due to the resource constraints of Fog Nodes which are resource constrained (performance limited). For example, if we have nth users connected to fog service which provisions the e-learning framework, this nth number of users will be divided over the number of fog nodes with respect to the user's location. While this may be an improvement over the cloud, it could become a challenge when there is a geometric increase in the number of user connections on a particular node. This can lead to a degradation in QoS and possibly system failure due to computational overhead [9][10][11]. However, the benefits of FC to real-time e-learning (low latency and reduce bandwidth consumption) cannot be neglected. Therefore, in this work, we propose a load balancing framework in geo-distributed fog computing environment designed for e-learning system. The load balancing framework is capable of managing users' connection, enhance CPU utilization, improve response time and network bandwidth management using the fog computing framework [12][13]. The contributions made in this paper include:

- Design a resource management recommendation system to manage user connections in the Fog Landscape.
- Develop analytical model for the problem
- Design an algorithm to determine the connection threshold of fog nodes
- Design an enhanced computing framework capable of optimizing CPU utilization, improved response time and network bandwidth management.

2. THE PROPOSED LOAD BALANCING FRAMEWORK

The proposed system, presented in Figure. 1, is based on the concepts of fog computing techniques. The general architecture is a collaboration of three hierarchical layers: the cloud layer (top layer), the fog layer (middle layer) and the user layer (bottom layer) [4][5][6]. The proposed system considered all users who wish to connect to the system in order to gain access to the e-learning platform via requests sent to the fog layer. In this scenario, the system considers request as a 'request for connection' to the proposed system. The fog layer, as stated earlier is the middle layer were most of the user activities are be handled. This layer comprises of intelligent devices such as routers, switches, access points, mini datacenters, SBCs etc) known as fog nodes. The fog layer has a finite number of fog nodes which are sparsely distributed. A fog node may be static or mobile as well as capable of accepting new user request.

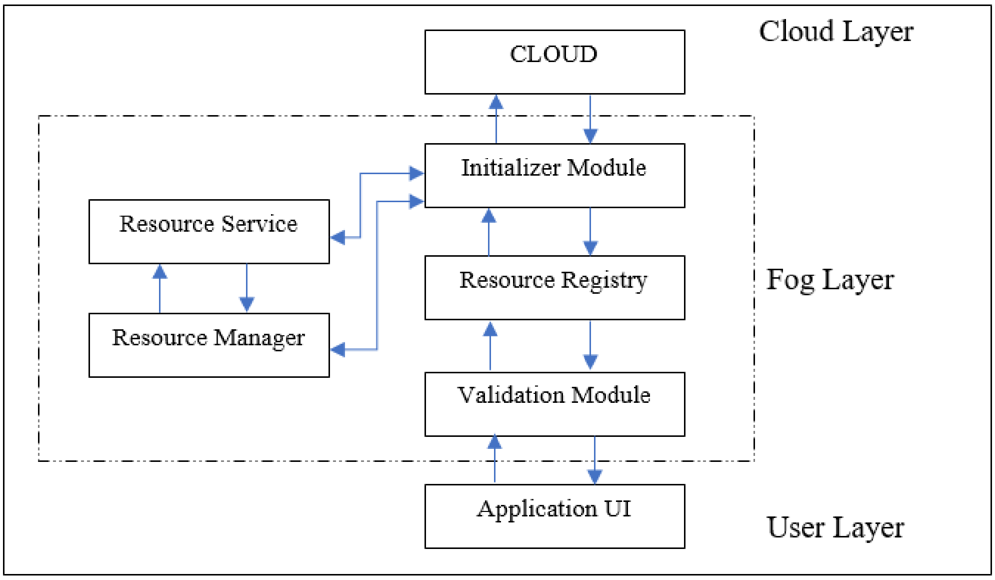

Figure 1. Layered architecture of the proposed frame work

An important part in the architecture of the proposed system is the c Fog Controller, which is embedded in all fog nodes in the system. The fog controller has three (3) major components (initializer, resource registry, resource manager and service manager), it is responsible for initializing the fog node as well as updating the fetching of new user registration details and new fog placement coordinates usually originating from the cloud; this process is event-driven. The information gotten is uploaded into the resource registry. This process is repeated whenever the fog node is initialized. During the initialization process, the resource manager and resource service are updated with information via the resource registry. The resource manager component incorporates a set of algorithms and policies that are used to manage the computing resources within the fog node. It communicates with the resource service as well as the resource registry to get update information and parameters needed for computing. The resource service receives and hold information such as resource configuration, network bandwidth, self-location coordinates, location distance of all neighbor nodes current active connections count and CPU status gotten from the resource registry and manager. This information is used to update neighbour nodes on the current status of the fog node and the information is useful for making the recommendation decisions.

3. MATHEMATICAL FORMULATION

The use case is presented in this section along with the mathematical model of the proposed framework that uses the fog computing architecture to solve the problem.

3.1 Problem Description (Use Case)

Consider a scenario where there are 1500 users who need to access the e-learning platform in a Fog Landscape with available fog nodes within a given location. Saieed et al. (2021) proposed a model which divides the sample size into small equal parts (or evenly distributed parts) and each part will connect to the nearest fog node since fog nodes by nature are decentralized. While this proved successful in their context and within the scope of their research, some other aspects of the scenario need to be considered. Imagine the possibility that the 1500 users are not evenly distributed within the fog nodes locations. This implies that some fog nodes will be under loaded while others over loaded. This can negatively impact on the response time, cause a system failure and possibly lead users back to the cloud. Within this context, a load balancing problem is identified which the proposed solution presented in this paper has been designed to solve.

Let C be a collection of fog nodes within a given coverage area with a given number of fog nodes given as Cn. Each collection has a location referenced fog node Mh with location (x,y) called origin node used to calculate the distance D between any 2 fog nodes within the collection. M represents a homogeneous set of fog nodes (m1,m2,…,mn) in a given collection with location ((x1,y1),(x2,y2),…,(xn,yn)). It is assumed that all location of fog nodes are determined during placement of the fog node into the collection and stored in the cloud for easy update by administrators and access by fog nodes. It is assumed that each fog node m ∈ C. Let U be a set of users (u1,u2,…,un) static or mobile and all users are registered into the e-learning platform via the cloud interface; this allows all fog nodes have access to this information for user validation. The connection threshold of fog nodes is given as T which is mostly static. Mcn is given as fog nodes current connection count. While Neighbouring which is defined as any fog node that is close to another fog node based on certain criteria is given as Nm, and its current connection count be given as Ncn. The signal coverage area of fog node is given as MR and the fog coverage area is given as Mss. The various parameters concerning the framework definition are presented in Table 1.

3.2 DEFINITION OF VARIABLES

Table 1. The various parameters concerning the framework definition

Symbol	Definition	Remark	Type-Unit
C	Collection of fog nodes	Index	-
Cn	Number of fog nodes in collection	variable	Integer – [unit]
M	Homogeneous set of fog nodes (m1,m2,…,mn)	Index	-
Mh	Origin fog node with Location coordinates (x,y)	Set	Integer – [unit]
D	Distance between 2 fog nodes	Calculate	Integer – [m]
U	Users accessing the e-learning platform	Index	-
T	Connection threshold of fog nodes	Set	Integer – [unit]
Mcn	Fog node current connection count	variable	Integer – [unit]
N	Neighbour nodes List determined by Mss distance	Index	Integer – [m]
Mss	Fog node signal coverage for neighbours	Calculate	Integer – [m]
Ncn	Neighbour current connection count	variable	Integer – [unit]
MR	Signal Range of Fog node	Set	Integer – [unit]
Rc	Rquest for Connection	index	-
Con	Successful Connection		

3.3 DEFINITION OF CONSTRAINTS

a. Constraint to satisfy a successful connection by a user.
This constraint is determined by the validation module which checks a user’s connection request (Rc) details against that stored in the resource registry.

Rc -> V = true; if validation is successful, accept connection.
Rc -> V = false; if validation is unsuccessful, reject connection.

b. Constraint to accept a validated connection request
This constraint is determined by two factors:

i. If neighbour node active connection count is less than source node connection count;
Ncn < Mcn;

ii. If distance (D) between neighbour node (y) and source node (x) is less than or equal to half the source node (MRs) signal range given as (Mss). To ensure user is always within range of fog nodes being recommended;

$$Mss = MRs/2 \quad (1)$$

Using Pythagorean theorem, D is given by:

$$D = \sqrt[2]{\left((x1 - x2)^2 + (y1 - y2)^2\right)} \quad (2)$$

D <= Mss must be satisfied for connection to be granted.

iii. If the neighbour requirement given in (ii) is not fulfilled, then source node will check if its own connection count (Mcn) gas reached its threshold (T) and connect or reject the connection request

Mcn < T;

4. PROPOSED FRAMEWORK DESIGN PHASES

The proposed system was designed with a modular concept for flexibility and easy scalability. In this section, a decomposition perspective with detailed flow process is presented.

4.1 FOG NODES INITIALIZATION PHASE

The initialization phase entails the configuration and setup of the fog nodes to receive user connections. When any fog node within the collection is powered, it updates its source registry with current information on the registered users, the current status of all fog nodes within the collection. This phase is required on every start up to ensure all fog nodes in the collection are updated with any change in fogs nodes list (new installation or removal) as well as users (new registered user or removal). Secondly, the information updated from the cloud is used by the fog node resource service to determine the neighbour nodes based on parameters and equations presented in section 3.3b(ii), creates a list of neighbour nodes from fog nodes that meets the stated criteria and stores same for exchange with requesting nodes. The resource service also collects information on current active connection from the resource manager and stores same for access by other fog nodes. After this process the Application User Interface (AUI) is loaded and prepared to receive connection request from users via an http request. The flow diagram and algorithm of the process are presented in Figure 2 and algorithm 1.

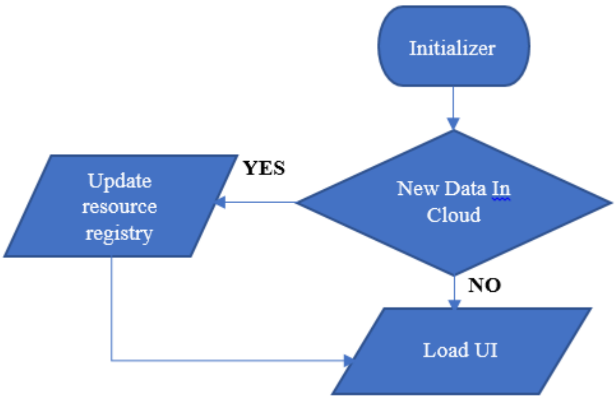

Figure 2: Fog Node Initialization Phase Flow Diagram

Algorithm 1. Initialization system
1: Function initialize_fogNodes()
2: //initializer connects to the cloud
3: Let newData Nd
4: if N = true //download new data and update resource registry
5: else //continue
6: //resource service fetch fog nodes location coordinates
7: //from resource registry and calculate distance from all other nodes to form Neighbour list
8: Let distance D
9: D = /(x1 – x2)2 + (y2 – y1)2
10: //calculate node signal coverage for neigbours
11: Let sourceNode Mss
12: Mss = MRs/2
13://create neighbour list
14: Let neighbourNode N
15: if D <= Mss, N= true //Add node details to neighbour list
16: else //continue
17: //get connection status info and connection count from neighbour nodes via resource managers
18: //Load Application user Interface
19: //wait for user connection
20: pause

4.2 USER CONNECTION PHASE

This phase is responsible for three (3) major steps, firstly validating users credentials against records stored in the fog nodes source registry, secondly accepting a user request on the source node and thirdly, using information from its resource registering and that of the neighbour nodes to recommend the right fog node for a user based on the constraints presented in section 3.3. The detail description of the user connection phase are shown in the flow diagram of Figure 3 and in algorithm 2.

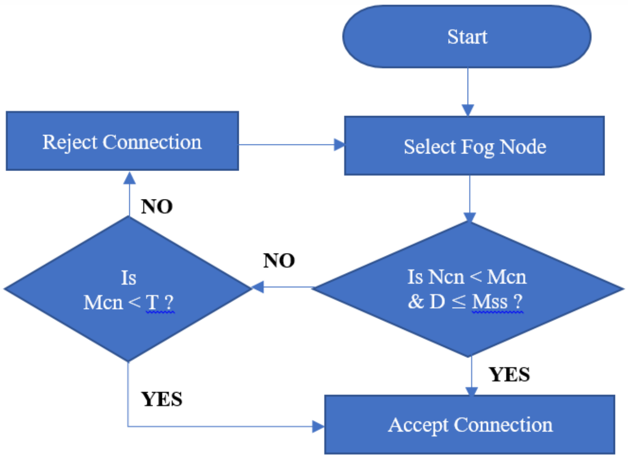

Figure 3: User connection Phase Flow Diagram

Algorithm 2. Proposed User Connection system
1: Function connect_Users()
2: //user select a fog node within range to connect to from list
3: Let fogNodes M = [m1,m2,…,mn]
4: //Input: user login details.
5: // Fetch User Login Details and Validate
6: Let V = validation
7: // validation is successful connect the user else reject connection
8: if V = true //proceed
9: Else V =false //reject

```
//fetch list of neighbour nodes determined mathematically to recommend proper placement
Let neighbourNodes N = [N1,N2,…,Nn]
//fetch connection count status of source and neighbour nodes and compare
Let sourceNode Mcn
Let neighbourNode Ncn
Let connectionThreshold T
Let successfulConnection Con
if Mcn < Ncn // check Mcn connection Threshold (T)
    if Mcn < T //accept connection on Mcn
Con = true //end connection process
    if Mcn > T //reject connection request
Con = false //end connection process
if Mcn > Ncn //accept connection and recommend connection to Ncn
Con = true //end connection process
Else Con =false //end connection process
end
```

5. CONCLUSION

This paper proposed a framework designed to enhance the management of user connections in a geo-distributed environment, to avoid overloading or underloading fog nodes that may lead to system failure due to computational overhead or underutilization of available compute resources. The proposed model is based on the fog computing architecture that enables a distributed setup thereby allowing for segmentation of users into smaller groups to prevent over subscription on a particular system. The distributed architecture of fog computing and the proposed model ensures that no one node is overloaded or underloaded in order to avoid a system failure. As presented in the flow diagram and algorithm of the proposed model, the users are connected based on a systematic recommendation process so as to connect to the most suitable node within the location of the user. This will improve the performance of the e-learning environment by having a more evenly distributed connection. Finally, the proposed framework allows for scalability with capability for future improvements and further research to develop a leading framework for connection automation.

REFERENCES

[1] K. B. Moldashev, G. Zh. Kuchumova, N. S. Malgeldinov, University Strategic Response to Pandemic Challenges: A Systematic Literature Review And The Case Of Kazakhstan, Central Asian Economic Review, 10.52821/2789-4401-2022-5-127-141, 5, (127-141), (2023).

[2] Putro, H. P. N., Hadi, S., Rajiani, I., & Abbas, E. W. (2023). 16. Adoption Of E-Learning In Indonesian Higher Education: Innovation or Irritation?. *Membelajarkan Diri Menulis Membukukan Tulisan*, *22*(1), 177.

[3] Al Rawashdeh, A. Z., Mohammed, E. Y., Al Arab, A. R., Alara, M., & Al-Rawashdeh, B. (2021). Advantages And Disadvantages of Using E-Learning In University Education: Analyzing Students' Perspectives. *Electronic Journal Of E-Learning*, *19*(3), 107-117.

[4] Sunyaev, A. (2020). Cloud Computing. In: Internet Computing. Springer, Cham. Https://Doi.Org/10.1007/978-3-030-34957-8_7

[5] Ibrahim, A.H.; Fayed, Z.T.; Faheem, H.M. Fog-Based Cdn Framework For Minimizing Latency Of Web Services Using Fog-Based Http Browser. Future Internet 2021, 13, 320. Https://Doi.Org/10.3390/Fi13120320

[6] Aleisa, M.A.; Abuhussein, A.; Alsubaei, F.S.; Sheldon, F.T. Examining The Performance Of Fog- Aided, Cloud-Centered Iot In A Real-World Environment. Sensors 2021, 21, 6950. Https://Doi.Org/ 10.3390/S21216950

[7] Saieed, M., Ahmed, H., & Abbas, N. (2021). A proposed learning model based on fog computing technology. *IJCI. International Journal of Computers and Information*, *8*(2), 137-143.

[8] Jameel, S. M., & Croock, M. S. (2020). Mobile learning architecture using fog computing and adaptive data streaming. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(5), 2454-2462.

[9] Alsadie, D. (2022). Resource Management Strategies in Fog Computing Environment-A Comprehensive Review. *International Journal of Computer Science & Network Security*, *22*(4), 310-328.

[10] H. Rafique, M. A. Shah, S. U. Islam, T. Maqsood, S. Khan And C. Maple, "A Novel Bio-Inspired Hybrid Algorithm (Nbiha) For Efficient Resource Management In Fog Computing," In Ieee Access, Vol. 7, Pp. 115760-115773, 2019, Doi: 10.1109/Access.2019.2924958.

[11] Ghobaei-Arani, M., Souri, A. & Rahmanian, A.A. Resource Management Approaches In Fog Computing: A Comprehensive Review. *J Grid Computing* **18**, 1–42 (2020). Https://Doi.Org/10.1007/S10723-019-09491-1

[12] T.-W. Kuo And A. Mok. Load Adjustment In Adaptive Real-Time Systems. In Proc. Ieee Real-Time Systems Symposium (Rtss), Dec. 2020.

[13] Matrouk, K., & Alatoun, K. (2021). Scheduling Algorithms In Fog Computing: A Survey. *Int. J. Networked Distributed Comput.*, *9*(1), 59-74.