CLOUD COMPUTING: A LATENCY AND BANDWIDTH COST OPTIMIZATION PERSPECTIVE

Himangi Agrawal, Tanya Gupta
E-Mail Id: 21bec046@nirmauni.ac.in, 21bec125@nirmauni.ac.in
Department of Electronics and Communication Engineering, Institute of Technology, Nirma University, Ahmedabad, India

Abstract- Cloud providers are forced to run numerous datacenters throughout the world to host their cloud services due to end-user latency and regulatory requirements. Request allocation, the process of allocating each user request to the best data center to benefit cloud providers, is an emerging issue under such geo-distributed architecture. Nevertheless, previous request allocation solutions have serious drawbacks: They either only optimize benefits for one party (providers or users, for example) or neglect some important but realistic factors (various per-unit bandwidth costs among datacenters and heterogeneous latency requirements of different users) when optimizing benefits for both parties. The challenge of ensuring end user's latency requirements while minimizing the overall bandwidth cost is discussed in this paper. In order to meet latency constraints and reduce operating costs, it is critical to optimize user request allocation as cloud services continue to expand across diverse geographical regions. The problem of effectively allocating incoming requests in a geo-distributed cloud environment is discussed in this paper. The principal aim is to achieve equilibrium between cost minimization and adherence to predetermined latency constraints, thereby guaranteeing responsive services for users, regardless of their geographical location. The suggested method makes use of intelligent load balancing, dynamic resource scaling, and predictive models to optimize inter-region data transfer while adaptively allocating requests. To assess resource provisioning and data transfer costs, cost models are integrated, which enhances overall cost-effectiveness. Metrics like reliability, scalability, cost efficiency, and latency adherence are used to assess how effective the system is. The findings are intended to contribute to a more responsive and economical cloud infrastructure by offering insights into improving the performance of geo-distributed cloud services.

Keywords: Bandwidth, Cloud performance, Latency, Random sampling, Request allocation, Security.

1. INTRODUCTION

The way that latency and bandwidth cost interact is a key factor in determining how well cloud services perform overall. User experience is directly impacted by latency, or the amount of time it takes for data to move from the user's device to the cloud server. High latency can negatively impact the perceived quality of services by causing delays in data retrieval and application responsiveness.

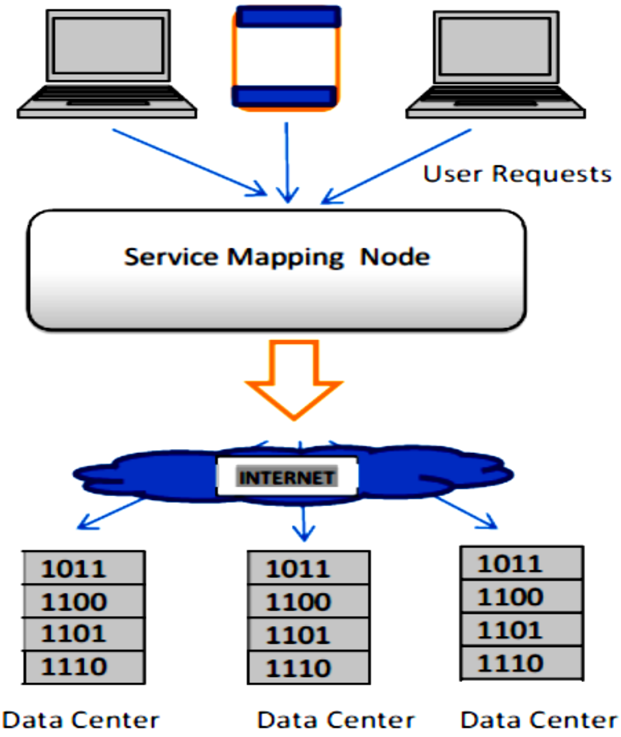


Fig. 1.1 User Request Allocation Diagram Datacenters [2]

However, a key element in the cost-effectiveness of cloud operations is bandwidth cost, which stands for the costs related to data transmission. Optimizing the economic aspects of cloud service delivery requires minimizing bandwidth costs, even though reducing latency is crucial for a responsive user experience. Finding the ideal ratio between these variables is a complex task.

The necessity for more effective and possibly expensive data transmission techniques could result in higher bandwidth costs if latency is aggressively reduced [7]. On the other hand, concentrating only on reducing bandwidth expenses could jeopardize latency and, as a result user satisfaction. So, attaining an optimal solution of cloud performance needs a thorough strategy that takes into account the complex relationship between latency and bandwidth cost, with the goal of delivering data efficiently and ensuring a seamless user experience. The problem of request allocation is important, but it also has two challenges. Initially, it is common for multiple requests to be made at the same time by various users, each of whom may have varying latency requirements. Furthermore, a datacenter's uplink may become congested due to an excessive number of requests, which could cause long queuing times and possibly satisfy end users' latency requirements. Secondly, cloud service providers usually spend a lot of money renting Internet service providers' bandwidth for traffic to and from their data centers. Additionally, because datacenters are dispersed throughout different regions, a cloud service provider may rent bandwidth from multiple ISPs with a range of pricing structures, creating a notable degree of heterogeneity in the cost of bandwidth per unit across data centers [5]. To the best of our knowledge, though, the aforementioned difficulties with the request allocation problem for geographically dispersed cloud services cannot be resolved by current work. This paper addresses the fundamental challenge of efficiently allocating concurrent user request to geographically distributed data centers within a cloud service provider’s network.

2. MOTIVATION

Certain existing proposals only aim to minimize operating costs of service providers, while others merely ensure that end users receive high quality services. However, some solutions merely ignore the variation in latency from various end users and the variation in per unit bandwidth cost of various data centers, even though they do optimize the benefits of both service providers and end users. As a result, these programs have very little effect on the real situation in terms of ensuring that end users receive guaranteed performance while service providers incur the least amount of bandwidth. It is imperative to optimize latency and bandwidth expenses to guarantee the smooth operation of these services. Saving money for end users and service providers alike is facilitated by optimizing latency and bandwidth usage. Workers in cloud environments are not all the same. Adaptive scaling, load balancing, and efficient resource provisioning are critical for dynamically satisfying demand while controlling bandwidth expenses.

3. LITERATURE REVIEW

Although the effect of security on network infrastructure has been extensively demonstrated by various researchers, the effect of security on cloud performance may appear a little odd. One such example is the widespread effects of distributed denial of service (DDoS) attacks on network performance[1]. These attacks not only compromise response times but also pose a threat to security. Cloud performance is greatly improved by virtual machine migration since it distributes the load among multiple data centers. It was a sophisticated form of migration that offers strong and quick response in data centers.

When physical hardware is abstracted into virtual instances through virtualization, more processing power is needed to manage and maintain these virtual machines. This abstraction's overhead may result in a less economical use of computing resources by affecting system performance and resource utilization overall. Additionally, issues like worsening network congestion and a higher chance of service interruptions could arise from the complexity and overhead brought about by the virtualization layer. The instability and dependability of the system may be impacted by the frequent migrations and adjustments of virtual instances made to reduce latency. To further complicate the search for the best latency reduction, the reliance on real-time monitoring for efficient virtual machine placement raises questions about prediction accuracy and potential delays [2]. Consequently, even though virtualization has many advantages, a balanced and successful deployment depends on careful evaluation and management of the disadvantages it entails.

Table-3.1 An Overview of Literature Review

Author	Year of Publication	Parameter	Scheme	Pros	Cons
Malvinder Singh Bali et. al.	2013	Cloud computing, performance	Effect of latency on domains of cloud network and service disruption due to DDoS attack on cloud network.	1. Unlimited storage space with less time	1. Latency issues 2. Disruption in services
Sonam Srivastva et al.	2016	Latency sensitive, and latency, virtualization	Different approaches to reduce latencyfor performance optimization.	Users' software programs are servers accessible. In caseof power outage, the program remains accessible to others.	User number exceeds the rated capacity adversely affecting the services.

Heng QI, Xinping XU	2020	Request allocation, latency, bandwidthcost	Reducing cloud provider bandwidthcosts and meeting end-user latency requirements.	Globally distributed data centers improve application reliability while decreasing user access delays.	Wastage of bandwidth resources. Expensive
Shin Jer Yang, Chuan-Hsin Chou	2017	Dynamic virtualized bandwidth allocation	Optimize network resource allocation to enhance virtual bandwidth allocation Improved network processing	Loosely connected control platform and data plane for centralized control. Improved QoS and resource optimization.	Poor network processing Inability to adjust bandwidth resources in real time for users.
Zhitao Wan	2010	Latency Sensitive	Virtualization, scalability, interoperability, quality of service, security, failure recovery.	Lower implementation andmaintenance expenses. Greater global workforcemobility Scalable, flexible infrastructure. Fast market entry	Cloud mesh is challenging to determine because toits dynamic shrinkingand expanding.

4. MATHEMATICAL MODEL

A mathematical model is discussed to analyze the issue of ensuring that user requests arrive at the requested latency while minimizing bandwidth costs for providers. The definition and justification of the symbolic variables are provided in Table 2. The presumptions and oversimplifications used in this paper are listed below. It is assumed that each datacenter houses the services and materials needed for every request, as is common amongst cloud service providers [3]. When it comes to content distribution networks (CDNs), the cache devices that are in charge of fulfilling user requests for content services are placed at the network's physical edge. If the edge layer is unable to fulfill its request, it will make one to the central layer, which in the worst situation scenario has to return to the source station. Additionally, we presume that a request's latency is divided into two components. The first component is the delay in transiting from user requests to the service gateway. For the optimization, the goal of Equation (1) is to reduce the cloud service provider's overall bandwidth costs. According to Equation (2), the total bandwidth used on the datacenter DJ's upstream link cannot be more than the matching bandwidth capacity ui. According to Equation (3), each request's latency must be limited by its requirement lj. The response time is measured by the first term, $\sum Pi\ (\sum rj\ bjxji)$, and the transport delay is represented by $\sum di\ \boldsymbol{hj}ixji$. Each request is assigned to just 1 datacenter, according to Equation (4). Lastly, Equation (5) ensures that xji can only accept 0 or 1. Where x_{ji} is not a 0-1 integer, but rather a continuous variable.

Table-4.1 List of Symbolic Variables

Symbol	Definition
M	Set of datacenters
d_i	Datacenter d_i which belongs to M
u_i	Uplink bandwidth capacityof particular datacenter
c_i	Per unit bandwidth cost of particular datacenter
N	Set of user requests
r_j	User request r_j whichbelongs to N
l_j	Latency requirement of userrequest
b_j	Bandwidth requirement of user request
hji	Transport delay between d_i and r_j
xji	Whether datacenter d_i serves request r_j
P_i	Response time of datacenter $\mathrm{d_i}$

$$\min \sum_{d_i \in M} \sum_{r_j \in N} b_j c_i x_{ji} \tag{1}$$

$$s.t. \sum_{r_j \in N} b_j c_i x_{ji} \leq u_i , \forall d_i \in M \tag{2}$$

$$\sum_{d_i \in M} P_i \left(\sum_{r_j \in N} b_j x_{ji} \right) x_{ji} + \sum_{d_i \in M} h_{ji} x_{ji} \leq l_j , \forall r_j \in N \tag{3}$$

$$\sum_{d_i \in M} x_{ji} = 1, \forall r_j \in N \tag{4}$$

$$x_{ji} \in \{0,1\}, \forall d_i \in M , \forall r_j \in N \tag{5}$$

5. ALGORITHM DESIGN

First step of the algorithm is to begin with the best possible solution for the given problem. After that, it selects a datacenter independently for every request in the for loop (Steps 2–10). In particular, it samples one datacenter with a probability of x_{ji} for every request rj. In the event that the sampled datacenter's bandwidth is insufficient or the user request exceeds the latency requirement, algorithm will continue sampling until a suitable datacenter is located. The algorithm has a number of noteworthy advantages. Initially, our algorithm's primary overheads are random sampling and solving the pertinent convex optimization. Standard solvers CVX can effectively solve the convex problem, returning the solution in 200 iterations for large-scale problems. Random sampling, on the other hand, only has an ON time complexity. Therefore, we draw the conclusion that our algorithm has very little overhead. Secondly, it would be simple to expand our algorithm to accommodate online use cases.

Algorithm for latency optimized request allocation

```
Input:
The amount of bandwidth to handle request $r_j$: {bj};
The latency requirement of request $r_j$ ;{lj};
The transport delay of request $r_j$ to data center di :{$h_{ji}$};
The uplink bandwidth capacity of datacenter di : {$u_i$};
Per unit bandwidth cost of data center di : {$c_i$};
Output:
Whether datacenter $d_i$ serves request $r_j$;
1.  Calculate the optimal solution and obtain {$x_{ji}$} where variables may be fractional;
2.  for each user request $r_j$ ∊ N do
3.      Sample one $d_i$ from the set M with a probability $x_{ji}$;
4.      If $u_i$ less than $b_j$ then
5.          Repeat step 3 ;
6.      else
7.          set $x_{ji}$ =1;
8.  Update the bandwidth capacity i.e. $u_i = u_i - b_j$ and the
    response time of data center $d_i$ i.e.$P_i$(.);
9.      endif
10. endfor
```

6. PERFORMANCE EVALUATION AND RESULTS

In the present paper cloud service provider with 40 datacenters is simulated. In the simulation, the units of all parameters are removed. In particular, each datacenter's downstream link's bandwidth capacity is set to 1000. Every datacenter's per-unit bandwidth cost is chosen at random from a range of [0.03, 0.3]. It is considered that 1000, 1500, and 2000 are concurrent user requests in simulation, with a latency requirement of between 50 and 500 for each request. Additionally, each request's required bandwidth is uniformly selected from a range of [5, 15]. Each datacenter's response time, or Pi, is calculated by multiplying its remaining capacity by a coefficient that is initially chosen at random from 1 to 100. The algorithm [11] is evaluated against two approaches. The latency-only algorithm is the first one; it assigns each request indiscriminately to the data center with the lowest total latency. The second algorithm is the cost-only one, which routes each request to the data center that charges the least for bandwidth.

In fig. 6.1, “LC” is referred as the suggested algorithm, ”LO” is referred as the latency-only strategy[13] that ignore the cost, and “CO” is referred as the cost-only strategy that ignore the latency, separately. The figure also illustrates how much bandwidth each algorithm will use overall for separate concurrent requests of 1000, 1500 and 2000. As it can be seen and in line with expectations, the total cost of LC is substantially lower than that of LO. Since LO only concentrates on latency optimization for individual requests, CO only minimizes bandwidth costs for data centers. In particular, the total cost of LC is lowered by 30% in 2000 concurrent requests when compared to LO and this can also meet the latency requirements for every user request. The outcomes clearly demonstrate how, when user requests are distributed among geographically dispersed data centers, the given algorithm can dramatically lower the overall bandwidth cost. User requests are highly sensitive to the experienced latency and although the LC algorithm doesn’t unilaterally minimize latency, it successfully meets the latency requirements of all requests. As demonstrated in table 3. The table represents the percentage of user requests with satisfied latency requirements under different algorithm types. Notably, both LO and LC ensure the latency requirements under different algorithm types and ensure the latency demands of all user requests , but CO falls short in this regard. This discrepancy costs without considering the latency requirements of user requests.

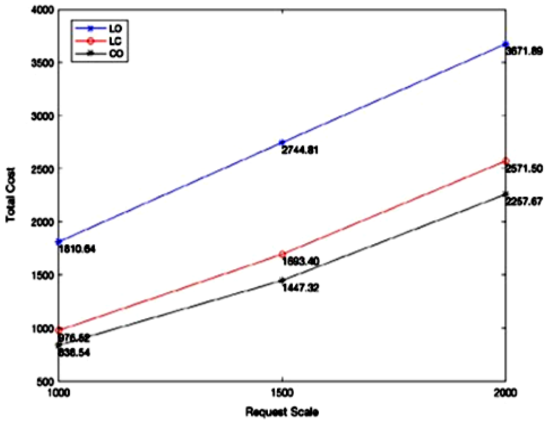
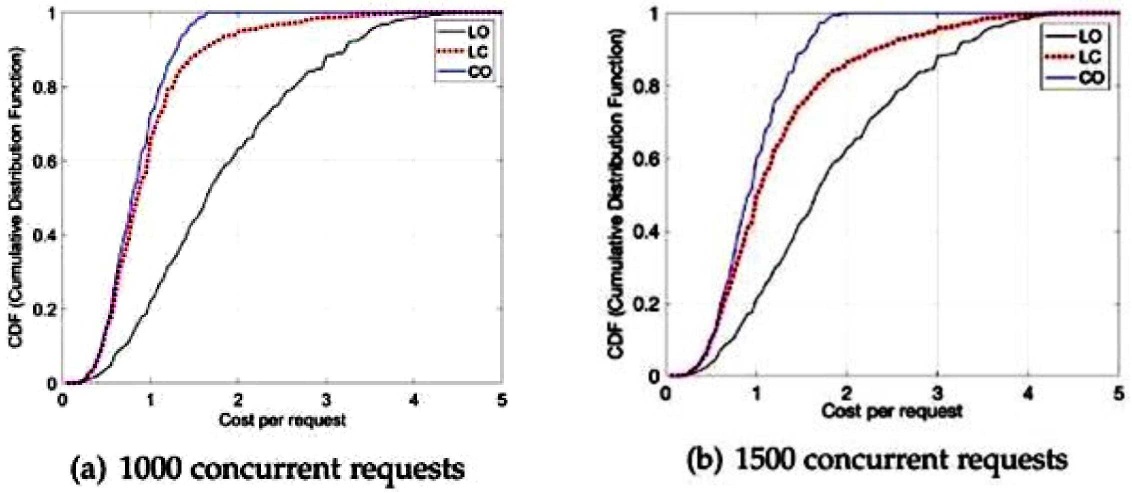


Fig. 6.1 Total Bandwidth Costs by Various Methods

Table-6.1 Rate of User Requests with Satisfied Latency Requirements

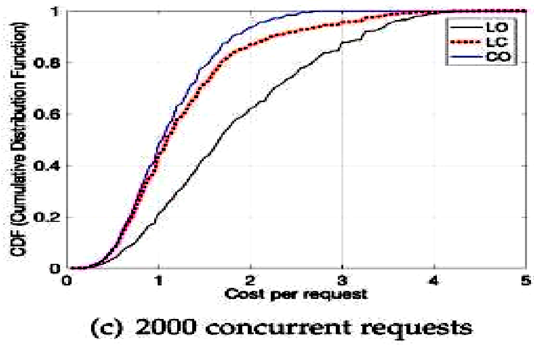
Algorithm	LO	LC	CO
Concurrent 1000	100%	100%	96%
Concurrent 1500	100%	100%	96%
Concurrent 2000	100%	100%	97%

In fig. 6.2, cumulative distribution functions (CDF) depict the bandwidth cost per request at 1000 and 2000 concurrencies, offering a detailed view of cost performance at a micro level. The LC curve consistently positions to the left of CO curve and generally approaches LO in most scenarios [12]. This indicates that the LC algorithm proposed in this paper consistently achieves lower bandwidth costs. Taking figure 3(c) as an example, further analysis reveals that, in this particular scenario, 80% of LC user requests correspond to a lower cost of 1.7. This signifies a cost reduction of 37% compared to LO algorithm and only a slight cost increase of 9% compared to the CO, which was unable to fully meet the latency requirement of all user requests.



(a) 1000 concurrent requests

(b) 1500 concurrent requests



(c) 2000 concurrent requests

Fig. 6.2 CDF Per User Request Cost

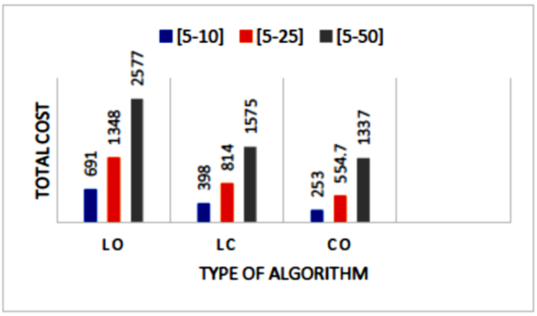


Fig. 6.3 Overall Cost For 500 Concurrent Requests Keeping the Latency Requirement Lj Fixed At 275

The bandwidth requirements for each request are randomly chosen from the range of [5, 15]. A question arises about the potential impact on the performance of the suggested request allocation algorithm when selecting bandwidth requirements within a different range. In each experiment, the range for randomly selecting bandwidth requirements is varied, while the latency requirement for each request remains fixed at 275. Fig. 4 illustrates the total bandwidth cost under different ranges of bandwidth requirement values. Notably, the LC algorithm consistently achieves relatively low bandwidth costs under various conditions [11-15]. Specifically , when altering the bandwidth requirement bi , from 5 to 10 , it is possible to reduce the total cost of LC by up to 42.5%in case of 500 concurrent requests compared to LO, all while meeting the latency requirements of each user request. These results clearly demonstrate that our technique remains efficient in reducing overall bandwidth costs even when bandwidth requirements change. Next, the influence of the latency requirements is examined. With bj maintained at a constant value of 25, the latency requirement 1_j is randomly selected in the intervals [50,100], [50,250], and [50,500], respectively. Fig 5 presents the results for a fixed bandwidth requirement of b_j =25. It closely resembles previous scenarios, showing a typical cost reduction of about 30% compared to LO. It’s worth noting that CO maintains a consistently lower value with varying latency requirements, but its delay requirement satisfaction increases from 70% to 96 % as the boundary is loosened.

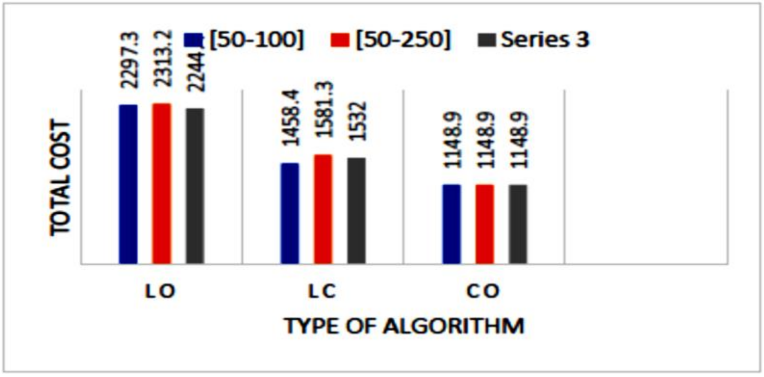


Fig. 6.4 Total Cost For 500 Concurrent Requests Keeping The Latency Requirement Lj Fixed At 25 (At Variable Bandwidth Requirements)

CONCLUSION

The newly-emerging issue of allocating each user request to the proper data center, with the goal of minimizing cloudservice provider’s overall bandwidth costs while maintaining end users' latency requirements is analyzed. After formulating an integer programming problem, a solution-friendly continuous convex optimization problem is analyzed. Next, a random sampling-based request allocation algorithm is observed to make sure that theoriginal optimization problem's solution is feasible, and asa result, the request allocation decision is determined. It is demonstrated that the ability of the algorithm to give a precise upper bound on the overall bandwidth cost. The results and graphs demonstrate that, in comparison to traditional algorithms, the described algorithm is more affordable for cloud service providers while still meeting end user latency requirements. These latency- reduction solutions are essential for high availability and fast reaction times in a variety of cloud-based applicationsas cloud computing evolves. The factors on which cloud performance rely have also been analyzed. The algorithm which takes into consideration both latency and bandwidth cost has been discussed. We finally conclude that if we focus only on reducing latency then the cost of users are increased and if cost id reduced then the latency requirements of users are not met. So there should be a trade-off between the two.

ACKNOWLEDGMENT

The authors would like to express our sincere gratitude to all colleagues who have contributed in successfully completing this paper. This endeavor would not have been possible without the unwavering support and guidance of Dr. Vivek Kumar Prasad, mentor and Professor in Nirma University, whose expertise and encouragement played a pivotal role in shaping the paper.

REFERENCES

[1] S. Srivastava and S. P. Singh, "A Survey on Latency Reduction Approaches for Performance Optimization in Cloud Computing," 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 2016, pp. 111-115, doi: 10.1109/CICT.2016.30.

[2] X. Xu, W. Li, H. Qi, J. Wang and K. Li, "Latency-Constrained Cost- Minimized Request Allocation for Geo-Distributed Cloud Services," in IEEE Open Journal of the Communications Society, vol. 1, pp. 125-132, 2020, doi: 10.1109/OJCOMS.2020.2964303.

[3] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, “Greening geographical load balancing,” in Proc. ACM SIGMETRICS, 2011, pp. 233–244.

[4] Z. Wan, "Cloud Computing infrastructure for latency sensitive applications," 2010 IEEE 12th International

[5] W.Li, X. Yuan, K. Li, H. Qi and X. Zhou, "Leveraging Endpoint Flexibility when Scheduling Coflows across Geo-distributed Datacenters," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, USA, 2018, pp. 873-881, doi: 10.1109/INFOCOM. 2018.8486319.

[6] P. Middleton, P. Kjeldsen and J. Tully, "Forecast: The Internet of Things worldwide 2013", pp. 57, 2013.

[7] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K.-R. Choo and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework", IEEE Access, vol. 5, pp. 8284-8300, 2017.

[8] M. Zahid, N. Javaid, K. Ansar, K. Hassan, M. K. Khan and M. Waqas, "Hill climbing load balancing algorithm on fog computing", Proc. Int. Conf. P2P Parallel Grid Cloud Internet Comput., pp. 238-251, 2018.

[9] X. He, Z. Ren, C. Shi and J. Fang, "A novel load balancing strategy of software-defined cloud/fog networking in the Internet of vehicles", China Commun., vol. 13, pp. 140-149, 2016.

[10] J. Kadhim and S. A. Hosseini Seno, "Maximizing the utilization of fog computing in Internet of vehicle using SDN", IEEE Commun. Lett., vol. 23, no. 1, pp. 140-143, Jan. 2019.

[11] R. Buyya and S. N. Srirama, Fog and Edge Computing: Principles and Paradigms, Hoboken, NJ, USA:Wiley, 2019.

[12] Y. Jiang, Z. Huang and D. H. K. Tsang, "Challenges and solutions in fog computing orchestration", IEEE Netw., vol. 32, no. 3, pp. 122-129, May 2018.

[13] Y. Liu, J. E. Fieldsend and G. Min, "A framework of fog computing: Architecture challenges and optimization", IEEE Access, vol. 5, pp. 25445-25454, 2017.

[14] S. V. et. al., “Life Extension of Transformer Mineral Oil Using AI-Based Strategy For Reduction Of Oxidative Products”, TURCOMAT, vol. 12, no. 11, pp. 264–271, May 2021.

[15] S. Yi, C. Li and Q. Li, "A survey of fog computing: Concepts applications and issues", Proc. Workshop Mobile Big Data, pp. 37-42, 2015.

[16] Vyas, M., Yadav, V.K., Vyas, S., Joshi, R.R. and Tirole, R. (2022). A Review of Algorithms for Control and Optimization for Energy Management of Hybrid Renewable Energy Systems. In Intelligent Renewable Energy Systems (eds N. Priyadarshi, A.K. Bhoi, S. Padmanaban, S. Balamurugan and J.B. Holm-Nielsen). https://doi.org/10.1002/9781119786306.ch5

[17] Sasanka Sekhor Sharma, RR Joshi, Raunak Jangid, Shripati Vyas, Bheru Das Vairagi, Megha Vyas., 2020, MITIGATION OF TRANSIENT OVER-VOLTAGES AND VFTO EFFECTS ON GAS INSULATED SUBSTATION. Solid State Technology, Volume 63, Issue 5/

[18] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, et al., "Security and privacy in fog computing: Challenges", IEEE Access, vol. 5, pp. 19293-19304, 2017.