QoS-Aware Fog Nodes Placement

Prasenjit Maiti*, Jaya Shukla†, Bibhudatta Sahoo‡ and Ashok Kumar Turuk§
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, Odisha
Email: *pmaiti1287@gmail.com, †jayashukla2192@gmail.com, ‡bibhudatta.sahoo@gmail.com, §akturuk@nitrkl.ac.in

***Abstract*—Fog computing has appeared as a favorable technology that can bring cloud applications closer to the physical IoT devices at the network edge but there is neither a common fog computing architecture or how it supports real-time Internet of Things (IoT) service execution. Edge devices such as the switch, router, gateway, mobile phones, smart car etc., are the candidates for deployment of fog nodes but the deployment differs according to the application. In this work, we have taken gateways as candidates for fog node deployment. The gateway collects data from smart sensors, but it does not have any pre-processing or decision-making capabilities. Therefore, the gateway is made smarter with Fog capabilities and named as Fog Smart Gateway (FSG). The processing of IoT traffic is taken care of by Virtual Machines (VMs) facilitated by distributed Fog nodes. We optimized the number of fog nodes for deployment to reduce the total latency induced by traffic aggregation and processing. Our results show that the optimal deployment of fog nodes in the IoT network could yield a reduction in latency compared to processing IoT data in a conventional cloud system.**

Keywords : Edge devices, Fog computing, Fog node, Service latency, Virtual Machines

I. Introduction

The Internet of Things (IoT) depicts a major change in data management. Real-time data management is associated with distributed objects and their associated smart sensors. Smart sensors data needs to be stored and retrieved efficiently on demand for IoT services. IoT devices are growing rapidly and it is anticipated that about 50 billion devices will be deployed in 2020. Existing cloud solution provides services for a large amount of data. But in some scenarios it can face limitations due to increased traffic of the entire network causing delay in processing the services. Different IoT services such as Health-care, Face Recognition, Military, disaster management require real-time response with very low latency. To overcome this problem, a new architecture needs to be proposed and thus, fog computing emerged to take care of these challenges. Fog computing is a concept that provides services at the network edge and involves smart Gateways named Fog Smart Gateways(FSG). Fog nodes are deployed in the network near the users to handle the services. In this architecture the data is processed locally before sending it to the cloud. The major issues and challenges of architecture design for edge-centric IoT services are discovering fog nodes, data caching, partitioning. Fog computing or Fog networking is a highly virtualized architecture for computing, storage, control and networking that distributes these services closer to end users and traditional cloud system. This platform supports micro-service delivery with reduced latency; bandwidth and network load for resource-constrained devices while maintaining service resiliency and localization. It has the potential to offer delay sensitive services for applications. It also supports added security, scalability, density of devices and mobility. IoT edge devices increase the computing resources to perform big data analytics. Fog node is a functional and conceptual entity in fog computing. A fog node is a physical device where fog computing is deployed. Fog node can provide infrastructure for IoT services execution. The common characteristics of fog node are that it is distributed and heterogeneous in nature, volatile, highly mobile and supports embedded computing, storage and networking capabilities for easy deployment of IoT applications or services. Most of the authors have not discussed implementation strategy of fog node.

II. Related Works

A fog network consists of several fog nodes, and each fog node resides in a base station or an access point such as switch, gateway, router etc. We effectively need a system that can efficiently select the set of edge devices based on functionality and characteristics of fog node to build a fog network. With virtualization technologies, a fog node has the capacity to run multiple virtual machines (VMs) on its own physical machine simultaneously, and a VM can be duplicated into multiple copies and placed in multiple fog nodes [1]. The VM can be flexibly placed in fog network, based on the traffic distribution and moving pattern of mobile user [2], [3]. However, the dynamic VM placement in fog node incurs a significant cost on latency and bandwidth consumption of the network links. The reason is, when a mobile user is in the coverage area of a fog node which has the application VM requested by the user, the service can be provided by the fog node without consuming any backhaul bandwidth. On the other hand, if the fog node does not host the application VM, to serve the user, extra backhaul data traffic will be generated by one of the following two cases: (i) the corresponding VM will be copied from some fog node owning the VM to the fog node in which the user resides through the backhaul network. In this case, the general traffic is termed VM traffic, and the amount of consumed backhaul bandwidth is related to the size of the application VM (ii) the user can directly access the application via the backhaul network to another fog node which has the VM for the application requested by the

user. In this situation, the bandwidth of backhaul network is consumed by the users access for the application service, and the generated traffic is termed data transmission traffic. We can observe that frequent VM replacement will cause serious VM traffic while static VM placement may result in significant data transmission traffic [4], [5]. Chen et al. [6] addressed the problem of video streaming service latency. Tziritas et al. [7] focused on the performance enhancement of cloud system using process migration and discussed experiment results with 1000 process. In another work, Chandio et al. [8] schedule 22,385 jobs to improve QoS. In IoT concern, the number is too low to be considered. In the above works discussed, the DCNs are serving the request of an application every time. Therefore, the DCNs are unable to process increasing number of IoT consumers requests within the DCs in real-time. The smart gateway is proposed as a fog node in [9], [10], the micro data centers proposed in [11], or the proposal of fog nodes serving as caches in Information Centric Networking in [12]. Fog nodes as mini-clouds proposed by [14], [15]. Now, the key aspect is where the fog nodes are located. Some of the authors [9], [10], [11], [14] proposed to locate fog nodes in highly capable devices, such as routers or smart gateways. Bonomi et al. [15], [16] proposed intermediate compute nodes as fog nodes which has no dependency on specific devices. Abdullahi et al. [12] and Skala et al. [17] proposed routers as a candidate for deployment of fog nodes. In the Fog to Cloud (F2C) scenario, whenever a node in the IoT layer submits a job , the node first contacts the fog it is connected with, instead of submitting the job directly to the cloud. The fog then decides whether to undertake the job itself or forward it to the cloud for processing. The fog has lower storage and processing capabilities than the cloud, it might at times refuse to undertake new jobs if it had reached its maximum capacity, hence the concept of fog availability is introduced in [18]. In this paper author discussed that fogs are not limited anymore to either execute a task or forward it to the cloud, but also to communicate with other fogs to process the job request. This minimizes the overall network delay. A management system responsible for discovering a set of available fogs is developed to select the best fog available to meet certain service requirements but authors have not explained the criteria used by the management system in choosing the optimal fog. Souza et al. [19] proposed a QoS- aware service distribution strategy in Fog-to-Cloud scenarios. The work aims at achieving low delay on service allocation by using service atomization in which services are decomposed into distinct subservices called atomic services to enable parallel execution. Although there exists extensive research in the area of cloud resource sharing; work in fog resource sharing and cooperation is still premature. It is quite natural that the service latency is drastically reduced as compared to the service processing using the cloud when fog computing is applied, but it is possible to further reduce the service response time depending on which nodes is deployed as fog nodes. Also, the network traffic can be reduced according to the deployment position of the fog node. Most of the research issues in fog computing

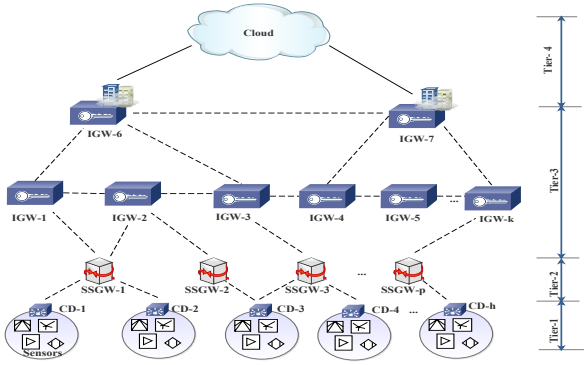

Fig. 1. QoS-Aware Fog Computing Architecture for IoT Services

[20], [21], [22], the service latency , network traffic and power consumption are reduced by fog computing architecture. But it is conceptual to deploy the fog nodes near the user only, and there is no consideration what devices actually fog server should be deployed on. Luan et al. [23] and Hong et al. [24] described the concept of fog computing as mobile fog and showed that mobile users can use fog computing to improve QoS, reduce bandwidth and energy consumption, end to end delay and network traffic. But, there is no consideration where to deploy or place the fog nodes. IoT gateway is an edge device which acquires data at the edge and normalizes and filters out that data. Intelligent IoT gateway makes decision locally and sends real-time service to the application. So, the IoT, smart gateway is the key component to collect data from various smart sensors node. As a consequence, real-time and latency-sensitive computation service requests to be responded by the distant cloud centers often endure large round-trip delay, network congestion, service quality degradation etc. To resolve these issues besides centralized cloud computing, a new concept named Edge computing or Fog computing has recently been proposed. Services are hosted at the edge of the network, and as a consequence, it reduces service latency, improves the quality of service (QoS) and provides a superior experience for end users [22], [25]. Several unique fog node architecture, application programming platform, mathematical model, and optimization technique have been proposed to attain certain Service Level Objectives (SLOs). Most of the attained SLOs are management oriented and cover latency, power, cost, resource, data, application, etc. related issues. We optimized the number of fog nodes to minimize the latency. A smart gateway is proposed to implement the so-called FSG supporting functions of resource estimation and management. We consider fog nodes location as a smart IoT gateway and optimally placed fog nodes.

III. Fog Computing Architecture

In our architecture (Figure 1) , the IoT service network consists of four layers. The networking elements of the architecture perform the tasks of data aggregation and processing of the traffic produced by IoT devices.

(A) Tier 1: This is the ground-level layer that includes all the smart sensor nodes (SSNs) that are assigned unique IPv6 addresses, suitably compressed according to the 6LoWPAN protocol and form a mesh network. SSN is a collection of sensors and actuators. These are responsible for sensing environment data and transmitting to its immediate upper layer. There can be instructions from the upper layer to the actuator to perform an action. SSN are distributed uniformly at random. A typical smart city scenario has hundreds of networks, pertaining the different domains, deployed all over its geographical area. Each of these networks is coordinated by a Coordinating Device (CD). A CD is known differently in different networks namely Cluster Head(CH) in sensor networks, Access Point(AP) in WiFi networks and Reader in Radio-Frequency Identification(RFID) network etc.

(B) Tier 2: CDs need to transmit their data to the Internet for efficient execution of their corresponding applications. This transmission of data is facilitated by the device known as Solution Specific Gateways (SSGW) or IoT Gateway (IGW). CDs can only communicate through one specific technology and are connected to at least one SSGW/IGW. However, an SSGW is a wireless device which supports technologies of all the CDs associated with it. Two SSGW to be connected if and only if they are in each others range and support at least one mutually common technology, else, they are connected through an IGW . SSGW s route the data received from CDs associated with them to the IGW s. The SSGW should also ensure the coverage of the CDs. Wireless Mesh Network is as close as it can get to the IoT network with one fundamental difference. All gateways in a wireless mesh network support the same set of technologies whereas SSGW in IoT supports different sets of technologies. Each IGW has a wired connection to the Internet and sends the data received from the SSGW s to the upper layer.

(C) Tier 3: This tier consists of edge devices such as switches, routers, access points, gateways. These devices temporarily store, process and analyze the received information. The fog computing devices support SSN mobility. FSGs receive data from CDs. All real-time analysis and latency- sensitive applications are run on the fog tier. Fog Nodes (FNs) are placed within IoT gateways specific to geographic locations. Each FSG serves multiple gateways within its proximity. The FSG is capable of load balancing, service management, resource provisioning of IoT gateways.

(D) Tier 4: The upper-most tier is cloud which is responsible for processing and storing an enormous amount of data to the high-end servers and data centers. A data center has several physical servers and there is an interconnection of high-speed LAN-network and high bandwidth link to the Internet from each physical server. Each IGW connected to a cloud data center by a wired network. The cloud computing environment is with the number of heterogeneous physical hosts in a data center.

IV. QoS Metrics

A. Service Latency

The service delay is the requested transmission delay and processing delay. We assume that the communication delay between SSNs is considered insignificant. Let δ_{cd_sg}, δ_{sg_igw}, δ_{igw_fsg} be the delays in transmission of a data packet from a CD to the corresponding SSGW, from a SSGW to the corresponding IGW, and from IGW to a fog smart gateway respectively. γ_{sg}, γ_{igw}, γ_{fsg} are the processing latency of SSGW, IGW and fog smart gateway for a data packet. Thus, the mean transmission latency, Φ_{fsg} , for the data packets of req_i request running within FN_i is given by

$$\Phi_{fsg} = \left(\delta_{cd\_sg}\alpha + \delta_{sg\_igw}\beta + \delta_{igw\_fsg}\theta\right) + \left(\gamma_{sg}\alpha + \gamma_{igw}\beta + \gamma_{fsg}\theta\right) \tag{1}$$

where, α, β, and θ $(\alpha > \beta > \theta)$ are the total number of packets sent by CD, SSGW, and IGW.

V. PROBLEM FORMULATION

We placed fog nodes in a gateway from where it can access maximum gateways data and delay is also minimized. We optimize the cost of the network using the minimal number of fog nodes. Each node transmits data to only one fog node. We assume that every gateway has decision-making capabilities. We model the IoT network as a graph $G(V, E)$ where V is the set of nodes (gateways in the network) and E is the set of undirected edges(link). Edge weights represents propagation latency, where $l(v, s)$ is the shortest path from node $v, s \in V$, and the number of nodes $n = |V|$. $S \subset V$ is a set of f number of fog nodes which are placed within gateways. The shortest path latency between each pair of nodes are stored in a distance matrix FDM and $\{FDM_{ij} \,|i, j \in n \;\; and \;\; FDM_{ii} = 0, FDM_{ij} = FDM_{ji}\}$. In the worst-case, if there is no limitation of fog nodes required to set up, the solution is to place a fog nodes at each gateway, but for the best case, the number of fog nodes should be restricted to $1 < f < n$. Hence our problem is to minimize the latency between gateways to fog nodes of the network. Selection of a gateway for fog node is represented by a binary selection variable Ψ_j, where $j = 1, 2, ..., f$.

$$\psi_j = \begin{cases} 1\,, if\ node\, v_j\, is\, selected\, for\, fog\, nodes\, placement \\ 0\,, \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \tag{2}$$

Let $L(S)$ represents the total latency between gateways to fog nodes.

$$L(S) = \sum_{j=1}^{f}\sum_{i=1}^{n} \min\ \left(l(v_i, \psi_j)\right) \tag{3}$$

$$L(S^{'}) = \min(L(S)) \tag{4}$$

Given the desired number of fog nodes f , there is a finite set of $\binom{n}{f}$ possible placements. The objective is to find the placement from the set of all possible fog nodes placement, such that the overall latency $L(S^{'})$ would be minimum.

VI. ALGORITHMS FOR FOG NODE PLACEMENT

This section discusses the details of proposed fog node placement algorithm. We are finding the appropriate mapping between gateways and fog nodes. We are applying K-means clustering with some modification to solve our problem. The algorithms find the f number of fog nodes considering distance as a metric between fog nodes to gateways. Our algorithm discusses the optimum arrangement of fog nodes into the selected gateways. The k-means clustering method gives different results for latency using different techniques for choosing the initial centroid. Six different techniques have been used for the selection of initial centroids and the final centroid of each cluster is chosen as the location for fog node placement. The Forgy method selects f random values from the n locations of the gateways. The Midpoint Method divides the set of gateways into f partitions and takes the midpoint of each of the f partitions as the initial centroid. The Sorted Cluster Midpoint Method first sorts all the locations as per the distance from origin and then calculates the midpoint of each partition as the initial centroid. The Partition Mean Method divides n gateways into f partitions and takes the mean of each partition as the initial centroid. The Sorted Partition Mean Method is similar to Partition Mean Method applied on sorted locations of the gateways. The Kauffman Method takes the mean as the centroid of first cluster. Then of all the gateways that are not selected the pairwise cluster distance is calculated for each gateway and the one that has maximum distance value with previously determined centroid is selected.

Algorithm 1: *Optimal_Fog_Nodes_Placement*

Input: FDM : nxn delay matrix of n number of gateways , f : number of fog nodes where $FDM \neq \Phi \wedge 1 < f < n$
Result: Location of the fog nodes

$Selection\_Initial\_Fog\_Nodes\,(FDM, f)$
while *not convergence* **do**
  for $i = 1 \;\; to \;\; n$ **do**
    Compute $membership(s_j \mid v_i) \;\; \forall membership \;\; (s_j|v_i) \in \{0, 1\}$
    $membership(s_j \mid v_i) = 1$; ▷ Delay between node v_i and centroid s_j is minimal
    $membership(s_j \mid v_i) = 0$ ▷ Otherwise
  end
  /* Recompute the center-gateway of these t clusters to find new cluster center-gateway s_j */
  for $i = 1 \;\; to \;\; n$ **do**
    for $j = 1 \;\; to \;\; f$ **do**
      $$s_j = \frac{\sum_{i=1}^{n} membership(s_j \mid v_i) v_i}{\sum_{i=1}^{n} membership(s_j|v_i)}$$
    end
  end
end

Algorithm 2: *Forgy method for selection of initial fog nodes*

Input: $Selection\_Initial\_Fog\_Nodes(FDM, f)$
Result: Initial location of the fog nodes

for $i = 1 \;\; to \;\; f$ **do**
  s_j = Select random v_i, where s_j is the centroid of cluster(j) and v_i is the gateway
end

Algorithm 3: *Mid Point method for selection of initial fog nodes*

Input: $Selection\_Initial\_Fog\_Nodes(FDM, f)$
Result: Initial location of the fog nodes

interval=n/f;
start=0;
finish=interval;
for $i = 1 \;\; to \;\; f$ **do**
  s_j = mid-point of $[v_i(start), v_i(end)]$ where s_j is the Centroid of Cluster(j) and v_i is the gateway
  start=finish+1
  finish=finish+interval
end

Algorithm 4: *Sorted Cluster Mid Point method for selection of initial fog nodes*

Input: $Selection\_Initial\_Fog\_Nodes(FDM, f)$
Result: Initial location of the fog nodes

for $i = 1 \;\; to \;\; n$ **do**
  $D(i)$ =Distance of gateway from centroid
end
Sort gateway on basis of distance from centroid
interval=n/f
start=0
finish=interval
for $i = 1 \;\; to \;\; f$ **do**
  s_j = mid-point of $[v_i(start), v_i(end)]$ where s_j is the Centroid of Cluster(j) and v_i is the gateway
  start=finish+1
  finish=finish+interval
end

Algorithm 5: *Partition Mean method for selection of initial fog nodes*

Input: $Selection\_Initial\_Fog\_Nodes(FDM, f)$
Result: Initial location of the fog nodes

interval=n/f;
start=0;
finish=interval;
for $i = 1 \;\; to \;\; f$ **do**
  s_j = mean of $[v_i(start), v_i(end)]$ where s_j is the Centroid of Cluster(j) and v_i is the gateway
  start=finish+1
  finish=finish+interval
end

VII. SIMULATION AND RESULTS

We have performed the simulation in the iFogSim simulator and run on the workstation equipped with Intel Core i7, 18 core processor, and 64 GB RAM. IoT gateways are assumed to be randomly distributed. We are fixing the number of gateways

Algorithm 6: *Sorted Partition Mean method for selection of initial fog nodes*

Input: $Selection\_Initial\_Fog\_Nodes(FDM, f)$
Result: Initial location of the fog nodes
for $i = 1$ *to* n **do**
  $D(i)$ =Distance of gateway from centroid
end
Sort gateway on basis of distance from centroid
interval=n/f
start=0
finish=interval
for $i = 1$ *to* f **do**
  s_j = mean of $[v_i(start), v_i(end)]$ where s_j is the Centroid of Cluster(j) and v_i is the gateway
  start=finish+1
  finish=finish+interval
end

Algorithm 7: *Sorted Partition Mean method for selection of initial fog nodes*

Input: $Selection\_Initial\_Fog\_Nodes(FDM, f)$
Result: Initial location of the fog nodes
v_1 = selected the most centrally located node as the first centroid
for $m = 2$ *to* f **do**
  for $i = 1$ *to* n **do**
    if v_i not selected
    for $j = 1$ *to* n **do**
      if v_j not selected
      Compute $C_{ij} = \max(D_j - d_{ji}, 0)$, where d_{ji} is the distance between v_i and v_j , D_j is the distance between v_j and nearest centroid.
    end
  end
  $s_j = v_i$ with maximum C_{ij}
end

to 32 to 512 and varying the number of fog node from 1 to 10. Data transfer from IGW to FSG in the form of the packet and the size of the packet are usually changed between 34 bytes to a maximum of 65550 bytes. The instruction size is 64 bits. Packet arrival is considered as a Poisson distribution with an average packet arrival rate of each node has 1 packet per second. Figure 2 to Figure 6 shows the latency (in milliseconds) and the number of fog nodes of the system for the optimal fog nodes placement algorithm. We analyze the service latency and the number of fog nodes. We observed that after placing 6 fog nodes the latency does not decrease that much. So, we can conclude that minimum 6 fog nodes required reducing the service latency. Random placement is far from optimal. Out of the six techniques Mid Point Method and Partition Mean Method show the best results.

VIII. Conclusion

It is observed that the service latency in fog computing environment are significantly lower than the cloud computing environment for a large number of real-time, low latency applications. In this work was detailed all the evolution in the planning and development of an architecture of fog computing

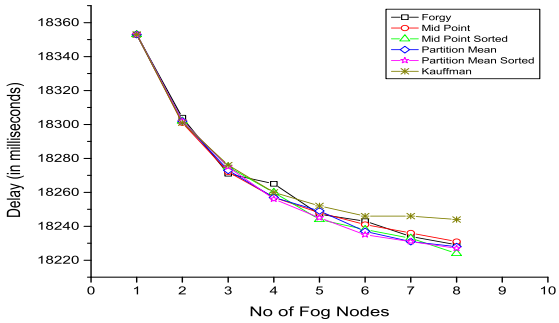

Fig. 2. Latency Vs. No of fog nodes for 32 gateways

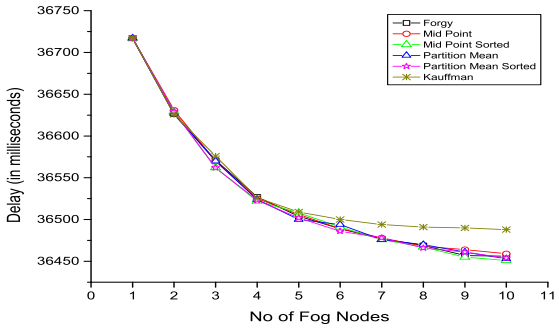

Fig. 3. Latency Vs. No of fog nodes for 64 gateways

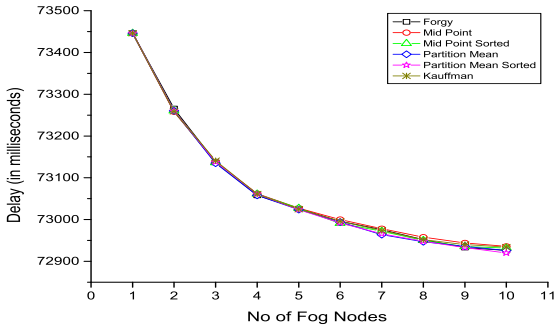

Fig. 4. Latency Vs. No of fog nodes for 128 gateways

for IoT services, having as main objective to efficiently bring together consumer and service providers. There are several complementary functions that fog are able to provide the user with a new generation of computing, and also serve as a requirement for real-time and low-latency applications to be the edges of the network.

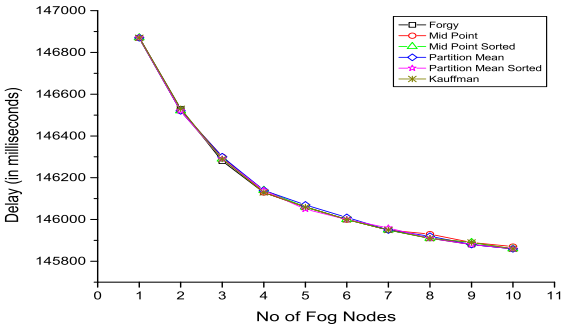

Fig. 5. Latency Vs. No of fog nodes for 256 gateways

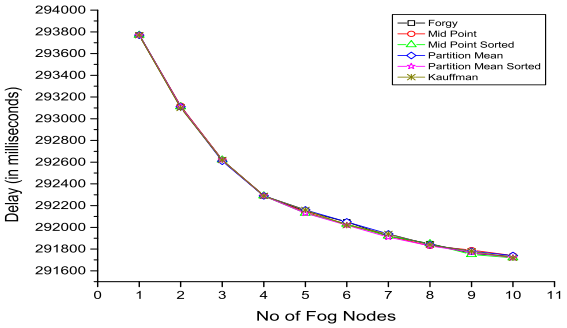

Fig. 6. Latency Vs. No of fog nodes for 512 gateways

ACKNOWLEDGMENT

This work has been supported by Media Lab Asia (Visvesvaraya Ph.D.Scheme for Electronics and IT, Project Code-CSVSE) under the department of MeitY government of India. The work is implemented at Department of Computer Science and Engineering, NIT Rourkela, India.

REFERENCES

[1] Y. J. Yu, T. C. Chiu, A. C. Pang, M. F. Chen and J. Liu, "Virtual machine placement for backhaul traffic minimization in fog radio access networks," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-7.

[2] G. Keller and H. Lutfiyya, Replication and Migration as Resource Management Mechanisms for Virtualized Environments, in Proc. Of ICAS, 2010, pp. 137143.

[3] V. Medina and J. M. Garc a, A Survey of Migration Mechanisms of Virtual Machines, ACM Comput. Surv., vol. 46, no. 3, Jan. 2014.

[4] X. Meng, V. Pappas, and L. Zhang, Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement, in Proc. of IEEE INFOCOM, 2010, pp. 11541162.

[5] T. Yapicioglu and S. Oktug, A Traffic-Aware Virtual Machine Placement Method for Cloud Data Centers, in Proc. of IEEE/ACM UCC, 2013, pp. 299301.

[6] W. Chen, J. Cao, and Y. Wan, QoS-aware virtual machine scheduling for video streaming services in multi-cloud, Tsinghua Science and Technology, vol. 18, no. 3, pp. 308 317, June 2013.

[7] N. Tziritas, S. U. Khan, C. Z. Xu, T. Loukopoulos, and S. Lalis, On minimizing the resource consumption of cloud applications using process migrations, Journal of Parallel and Distributed Computing, vol. 73, pp. 1690-1704, 2013.

[8] C. A. A. Chandio, K. Bilal, N. Tziritas, Z. Yu, Q. Jiang, S. U. Khan, and C. Z. Xu, A comparative study on resource allocation and energy efficient job scheduling strategies in large-scale parallel computing systems, Cluster Computing, Vol. 17, No. 4, 2014, pp. 1349-1367 .

[9] M. Aazam and E. N. Huh, "Fog Computing and Smart Gateway Based Communication for Cloud of Things," 2014 International Conference on Future Internet of Things and Cloud, Barcelona, 2014, pp. 464-470.

[10] T. N. Gia, M. Jiang, A. M. Rahmani, T. Westerlund, P. Liljeberg and H. Tenhunen, "Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction," 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, 2015, pp. 356-363.

[11] M. Aazam and E. N. Huh, "Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT," 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangiu, 2015, pp. 687-694.

[12] I. Abdullahi, S. Arif, S. Hassan, "Ubiquitous shift with information centric network caching using fog computing" in Computational Intelligence in Information Systems, Cham, Switzerland: Springer, pp. 327-335, 2015.

[13] K. Skala, D. Davidovic, E. Afgan, I. Sovic, Z. Sojat, "Scalable distributed computing hierarchy: Cloud fog and dew computing", Open Journal of Cloud Computing (OJCC), vol. 2, no. 1, pp. 16-24, 2015.

[14] A. M. Rahmani, N. K. Thanigaivelan , T. N. Gia, J. Granados, B. Negash, P. Liljeberg, H. Tenhunen "Smart e-Health Gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems," 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2015, pp. 826-834.

[15] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, "Fog computing and its role in the Internet of Things", Proc. 1st Edition MCC Workshop Mobile Cloud Comput., pp. 13-16, 2012.

[16] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics", Big Data and Internet of Things: A Roadmap for Smart Environments Springer Studies in Computational Intelligence, vol. 546, pp. 169-186, March 2014.

[17] K. Skala, D. Davidovic, E. Afgan, I. Sovic, Z. Sojat, "Scalable distributed computing hierarchy: Cloud fog and dewcomputing", Open Journal of Cloud Computing (OJCC), vol. 2, no. 1, pp. 16-24, 2015.

[18] W. Masri, I. A. Ridhawi, N. Mostafa and P. Pourghomi, "Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication," 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, 2017, pp. 1005-1010.

[19] V. B. Souza, X. Masip-Bruin, E. Marin-Tordera, W. Ramirez and S. Sanchez, "Towards Distributed Service Allocation in Fog-to-Cloud (F2C) Scenarios," 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, 2016, pp. 1-6.

[20] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang and R. Zimmermann, "Dynamic Urban Surveillance Video Stream Processing Using Fog Computing," 2016 IEEE Second International Conference on Multimedia Big Data (BigMM), Taipei, 2016, pp. 105-112.

[21] A. Brogi and S. Forti, "QoS-Aware Deployment of IoT Applications Through the Fog," in IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1185-1192, Oct. 2017.

[22] S. Sarkar, S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications", IET Netw., vol. 5, no. 2, pp. 23-29, Feb. 2016.

[23] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, L. Sun Fog computing: Focusing on mobile users at the edge. , 2015, arXiv preprint arXiv:1502.01815.

[24] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwlder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, ser. MCC'13. ACM, 2013, pp. 15-20.

[25] S. Sarkar, S. Chatterjee and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," in IEEE Transactions on Cloud Computing, vol. PP, no. 99, pp. 1-1.