Efficient Data Collection for IoT Services in Edge Computing Environment

Prasenjit Maiti
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, Odisha
Email: pmaiti1287@gmail.com

Jaya Shukla
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, Odisha
Email: jayashukla2192@gmail.com

Bibhudatta Sahoo
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, Odisha
Email: bibhudatta.sahoo@gmail.com

Ashok Kumar Turuk
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, Odisha
Email: akturuk@nitrkl.ac.in

***Abstract*—The Internet of Things (IoT) represents a major change in sensor data collection. It is predicted that 50 billion devices produce a large amount of data by 2020. That data needs to be stored efficiently so that it can be retrieved efficiently on demand for real-time application. Most of the Cloud-IoT solutions focusing on centralized data collection and storage which is not appropriate for efficient data collection and utilization. For addressing such diverse set of requirements, instead of sending all data to the Cloud, resources are placed near to the data sources for processing and fast real-time decision making. The gateway is such type of edge device that collects the data from smart sensors, but dont have any pre-processing or decision-making capabilities. Therefore, the gateway has to be made smarter with Fog capabilities and named as Fog Smart Gateway(FSG). We represent the distributed Cloud-IoT solution where optimally distribute data among mini-clouds/Fog nodes. The processing of IoT traffic is taken care of by Virtual Machines(VMs) facilitated by distributed mini-clouds/Fog nodes and located within the edge devices. We optimized the number of mini-clouds placement to reduce the total latency and power consumption induced by traffic aggregation and processing. To the best of our knowledge, this is the first work on mini-clouds placement. Our results show that the optimal distribution of mini-clouds in the IoT network could yield a total energy savings and latency reduced compared to processing IoT data in a conventional cloud system.**

Keywords: fog computing, fog node, IoT, edge devices, service latency, energy consumption

I. INTRODUCTION

The Internet of Things (IoT) [1] provides a smart environment for smart life. The smart environment such as smart transportation, smart health, smart building adopts sensor technologies, cloud technology. The typical solution for data storage, analysis, and processing is the cloud computing system which depends on Data Center Network (DCN). Data Centers(DCs) are processed all the requested services and resource demands in a cloud-based system. It is expected that a large amount of data will be generated by 50 billion devices on 2020 [2], [3]. Thus, by 2020, it is estimated that a large number of applications will be required to be processed and served through the technology of IoT. From 2007 to 2012 Barcelona started to develop the smart city with 1.62 million habitants in the area of $100km^2$ [4]. The city covered by the 320,925,019 sensors and produce 8GB(8583503168) data per day by the 150,000 lamposts, 40,000 garbage containers, and 80,000 public parking spots [5]. Data collection consists in combining the information generated in different sensors where the data is generated so that the amounts of data to be transferred can be reduced substantially. IoT applications demand real-time, low-latency services. To better appreciate the level of maturity of the enabling technologies for these services, we observe a synoptic view of the services in terms of suggested type(s) of network to be deployed; expected traffic generated by the service; maximum tolerable delay; device powering; and an estimate of the feasibility of each service with currently available technologies. The cloud DCs exhaust massive amount of energy leading to the emission of an enormous amount of greenhouse gases(GHGs), especially carbon dioxide(CO_2). This takes a deep toll on the environment. DCs is responsible for processing, storage, and computation of massive data generated from the IoT devices. The conventional cloud computing has a huge network bottleneck in terms of high service latency and poor Quality of Service(QoS). We analyze the suitability of recent computing paradigm- edge/fog computing [6] to serve the demands of real-time, latency sensitive applications(eg.mobile e-Health, smart traffic monitoring, smart parking etc.) in the context of IoT. Edge devices redirect the request to the cloud for permanent storage and historical data analysis. So, fog computing is not the replacement of cloud computing after investigating the different type of applications. The mini-cloud is responsible for reducing the congestion as well as balance the load in the network. It also provides services. Optimal positioning of mini-clouds to improve the QoS in terms of service latency, energy consumption, and cost efficiency. The focus of this work is optimally placed distributed fog nodes/mini-clouds for IoT-generated data with greater speed, closure to the point where the data was generated to reduce the latency and energy consumption.

We organized the rest of the paper as follows. Prerequisites for our work are in section-II as the literature review. Section-III describes the system model for edge-centric architecture for IoT services.Performance metrics described in Section-IV. Section-V represents the problem formulation and section-VI presents our proposed algorithm and its description. Simulation results from experimental assessment and observations are reported in Section-VII. We have concluded our work on this paper in Section-VIII.

II. LITERATURE REVIEW

IoT services need efficiently stored data so that it can access easily on demand. The IoT related services, including e-Health, smart cities, smart transportation systems and industrial scenarios are challenging the performance of cloud computing, mostly for the reasons of unpredictable and often high communication latency, privacy gaps and related traffic loads of networks connecting cloud computing to end-users. Over the last few years, researchers proposed many works [7],[8] ,[9], [10] on cloud services. In [11] Xiao et al. proposed the solution of design and optimal placement of DCs to improve the QoS in terms of service latency and cost efficiency. Chen et al. [12] addressed the problem of video streaming service latency. Tziritas et al. [13] focused on the performance enhancement of cloud system using process migration and discussed experiment results with 1000 process. However, the process migration within DCs is overhead degrading the performance for billions of processes in IoT. In another work, Chandio et al.[14] schedule 22,385 jobs to improve QoS. In IoT concern, the number is too low less to be considered. 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. IoT data can be classified into three categories (i) continuous data (ii) periodic data (iii) event-driven data. IoT gateway is an intermediate device between sensors and devices and the applications that create value from their data and access. The gateway allows you to efficiently collect and securely transport data from devices, remote users, and applications to serve a particular need. Today's industrial devices and other systems are often designed with inter-connectivity and the ability to share data. Intel, Dell, Huawei, Redhat, and AMD IoT Gateways enable companies to seamlessly interconnect industrial infrastructure devices and secure data flow between devices and the cloud. It also allows customers to securely aggregate, share, and filter data for analysis. It helps ensure federated data generated by devices and systems can travel securely and safely from the edge to the cloud and back without replacing existing infrastructure. 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. Fog nodes are able to collect and process a huge amount of data that are generated from sensors nodes through the gateways. A publish/subscribe model has been proposed in the Sensor-Fog-Cloud architecture for secure execution of services. The dynamic behaviors of the Sensor-Fog-Cloud infrastructure facilitate automatic furnishing of its services as required by the users. Fog computing environment supports easily gather, access and process and search for a large number of data by the mobile users. Bonomi et al. [6] stated that Fog computing is a highly virtualized platform that provides compute, storage and networking services between end devices and traditional cloud computing data centers. 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 [15] [16]. Resource and service management is one of the key challenges of Edge computing. In existing literature of fog computing many factors including time, energy, user-application context, etc. have been found playing important roles in resource and service management. In fog computing paradigm, time is considered as one of the important factors for efficient resource and service provisioning. 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. Therefore, edge computing offers an ideal placement for low-latency tasks, enabling the infrastructure to support emerging applications that demand real-time or predictable latency. Moreover, due to the capability to support a wide geographical distribution, edge computing is well positioned for big data aggregation, analyzing and distilling the bandwidth-hungry sensor data from devices. The data processing is now possible by the edge device itself, hence producing information pre-processed data, to be forwarded to the mini-cloud within the fog node. Fog-nodes are located at the edge devices. The processing of IoT traffic is handled by Virtual Machines (VMs) hosted by distributed mini clouds/fog nodes and located within the IoT networking elements. We optimized the number of mini clouds to minimize the latency of data collection. Optimally placed the mini-clouds and data-migration among mini-clouds is a research challenge. A smart gateway is proposed to implement the so-called fog micro-data center supporting functions of resource estimation and management. We consider mini-cloud location as a smart IoT gateway and optimally placed mini-clouds. We investigate, the tradeoff between power consumption and transmission delay in the

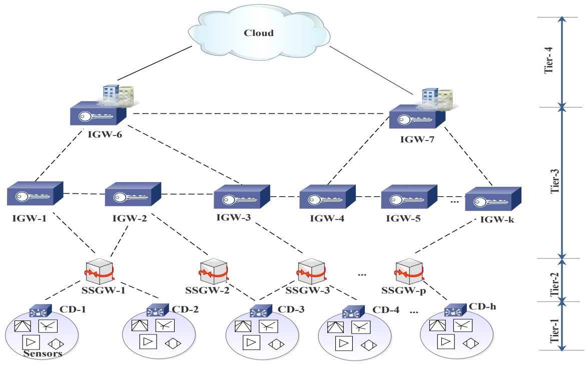

Fig. 1: Edge Centric Architecture for IoT Services

fog-cloud computing system. The workload allocation problem which suggests the optimal workload allocations between fog and cloud toward the minimal power consumption with the constrained service delay.

III. SYSTEM MODEL

In our model (Figure 1) , the IoT service network consisted of four tiers. The first tier(lowest) consisted of IoT devices and coordinators. The networking elements are located within the upper three layers. These networking elements perform the tasks of data aggregation and processing of the traffic produced by IoT devices.

(a) Tier 1: This is the bottom-most tier includes all the smart sensor nodes (SSNs) are assigned unique IPv6 addresses, suitably compressed according to the 6LoWPAN protocol and form a mesh network. SSN is a collection of sensor and actuator. 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. IoT devices or IoT nodes is a collection of SSN (mobile phones, smart vehicles, and smart meters etc.). SSN are distributed uniformly at random. A coordinate value is assigned to each SSN. We assume that the transmission range as a circle of SSN in a smart city scenario. 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 other's 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 IGWs. 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 gateway in a wireless mesh network supports the same set of technologies whereas $SSGW$ in IoT support 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 are temporarily stored, process and analyze the received information. The fog computing devices support SSN mobility. FSGs received data from CDs. All real-time analysis and latency- sensitive applications are run on the fog tier. Mini-Clouds(MCs) or Fog Instances(FIs) are placed within IoT gateways specific to geographic locations. Each FSG serve multiple gateways within its proximity. The FSG is capable of load balancing, service management, resource provisioning of IoT gateways. The service is the main motivation of the IoT. In general, a service is an autonomous, self-contained component capable of performing specific activities or functions independently, which accepts one or more requests and delivers one or more responses through a well-defined, standard interfaces. IoT services can represent as hardware devices, software resources, and any other thing or object that can be identified and located in specific places. Each IoT service can have many unique runnable instances with the same functionality but executed at different locations.

(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. PERFORMANCE METRICS

A. Service latency

The application instance sends a request for a service running on an IoT device. The service latency is response time which is calculated as the sum of the transmission latency and processing latency for a request. Let Δ_{cd}_sg and Δ_{sg_igw}, Δ_{igw_sfg} 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 smart fog gateway respectively. η_{sg}, η_{igw}, and η_{sfg} are the processing latency of SSGW, IGW and smart fog gateway for a data packet. Thus, the mean transmission latency, σ_{sfg}, for the data packets of req_i request running within mc_i is given by

$$\sigma_{sfg} = (\Delta_{cd\_sg}\mu + \Delta_{sg\_igw}\theta + \Delta_{igw\_sfg}\tau) + (\eta_{sg}\mu + \eta_{igw}\theta + \eta_{sfg}\tau) \tag{1}$$

where, μ , θ, and τ $(\mu > \theta > \tau)$ are the total number of packets sent by CD, SSGW , and IGW.

B. Energy consumption

The energy expended due to transmission of a unit byte of data from the CD to the SSGW, and from the SSGW to the IGW are denoted by λ_{cd_sg} and λ_{sg_igw}, respectively. λ_{igw_sfg} is the energy expanded of transmission of a unit byte of data from IGW to smart fog gateway. The energy required to process a unit byte of data within the SSGW, gateway, and smart fog gateway are denoted by ω_{sg}, ω_{igw}, and ω_{sfg} respectively. The rate of energy consumption for transmission and processing of data packets in fog computing environment is represented as

$$\xi(t) = \begin{pmatrix} \lambda_{cd\_sg} \sum\limits_{i=1}^{h} \sum\limits_{j=1}^{p} \gamma_{i,j} + \lambda_{sg\_igw} \sum\limits_{i=1}^{p} \sum\limits_{j=1}^{k} \alpha_{i,j} + \lambda_{igw\_sfg} \sum\limits_{i=1}^{k} \sum\limits_{j=1}^{t} \beta_{i,j} \end{pmatrix} + \begin{pmatrix} \omega_{sg} \sum\limits_{i=1}^{h} \sum\limits_{j=1}^{p} \gamma_{i,j} + \omega_{igw} \sum\limits_{i=1}^{p} \sum\limits_{j=1}^{k} \alpha_{i,j} + \omega_{sfg} \sum\limits_{i=1}^{k} \sum\limits_{j=1}^{t} \beta_{i,j} \end{pmatrix} \tag{2}$$

where $\gamma_{i,j}$, $\alpha_{i,j}$ and $\beta_{i,j}$ $(\gamma_{i,j} > \alpha_{i,j} > \beta_{i,j})$ be the total number of bytes being transmitted from cd_i to sg_j, sg_i to igw_j, and igw_i to sfg_j at time t.

V. PROBLEM FORMULATION

This section discusses the problem formulation of edge-centric data collection for IoT services. We geographically divide the smart city into sub-regions with their own surface road network. Each surface road connected together with one or more highway. Each sub-region and highway deployed vehicle sensors, weather sensors, road status sensors, smart watch sensors of the human body, sensors of body area network etc. The Large volume of data needs to be collected for monitoring and managing such system. Different entity (decision maker) of the smart city required a different type of data for decision making. All type of data may not be relevant to every entity. We stored each sub region's sensors generated data in a mini cloud and all mini cloud data stored in a cloud. We placed mini-clouds in a near to a gateway from where it can access maximum gateways data and delay also minimize. We optimize the cost of the network using the minimal number of mini-clouds. Each node transmits data to only one mini-clouds. 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 latencies , where $d(v, s)$ is the shortest path from node $v, s \in V$, and the number of nodes $n = |V|$. $S \subset V$ is a set of a k number of mini-clouds which are placed within gateways. The shortest path latency between each pair of nodes are stored in a distance matrix DM and $\{DM_{ij} | i, j \in n \;\; and \;\; DM_{ii} = 0, DM_{ij} = DM_{ji}\}$. In the worst-case, if there is no limitation of mini-clouds required to set up, the solution is to place a mini-clouds at each gateway, but for the best case, the number of mini-clouds should be restricted to $1 < k < n$. Hence our problem is to minimize the latencies between gateways to mini-clouds of the network. It is represented as an integer programming problem which is similar to 0/1 knapsack problem. So, it is a NP-complete problem. Selection of a gateway for mini-cloud is represented by a binary selection variable ψ_j, where $j = 1, 2, ..., k$.

$$\psi_j = \begin{cases} 1\,, if\, node\, v_j\, is\, selected\, for\, mini - cloud\, placement \\ 0\,, \hfill otherwise \end{cases} \tag{3}$$

Let $D(S)$ represents the total latency between gateways to mini-clouds.

$$D(S) = \sum_{j=1}^{k} \sum_{i=1}^{n} \min \;\; (d(v_i, \psi_j)) \tag{4}$$

$$D(S^{'}) = \min(D(S)) \tag{5}$$

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

VI. ALGORITHM FOR MINI-CLOUDS PLACEMENT PROBLEM

This section discusses the details of proposed mini-cloud placement algorithm. Here, the purpose is to minimize delay of data collection of the overall network. We are finding the appropriate mapping between gateways and mini-clouds. Clustering is a process in which a group of unlabeled patterns are partitioned into a set of clusters. The similar type of pattern is in one cluster. We are applying K-means clustering with some modification to solve our problem. The algorithms find the k number of mini-clouds considering distance as a metric between mini-clouds to gateways. Our algorithm discusses the optimum arrangement of mini-clouds into the selected gateways.

VII. SIMULATION AND RESULT

We have performed the simulation is in the iFogSim simulator and runs on the workstation equipped with Intel Core i7, 18 core processor, and 64 GB RAM. We consider a system with range $\{32, 64, 128, 256, 512\}$ mini-clouds or FI connected to a single CSP. IoT gateways are assumed to be randomly distributed. We are fixing the number of gateways to 32 to 512 and varying the number of mini-clouds from 1 to 9. 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 shows the latency (in milliseconds) and the number of mini-clouds of the system for the optimal mini-clouds placement algorithm. We analyze the service latency and the number of mini-clouds. We observed that after placing 6 mini-clouds the latency does not decrease that much. So, we can conclude that minimum 6 mini-clouds

Algorithm 1: $Optimal\_Mini - Clouds\_Placement(DM, k)$

```
Input: $DM$ : nxn delay matrix of n number of
       gateways , k : number of mini- cloud where
       $DM \neq \Phi \wedge 1 < k < n$
Result: Location of the mini-clouds
Randomly select the initial k number of gateways for
  mini-clouds placement
while not convergence do
    for $i = 1$ to $n$ do
        Compute
          $membership(s_j \mid v_i)$ $\forall membership$ $(s_j|v_i) \in$
          $\{0, 1\}$   ▷ if the node $v_i$ closest to the cluster
          $mc_j$(i.e the delay between node $v_i$ and centroid
          $s_j$ is minimal )then $membership(s_j \mid v_i) = 1$;
          otherwise $membership(s_j \mid v_i) = 0$
    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 $k$ 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
```

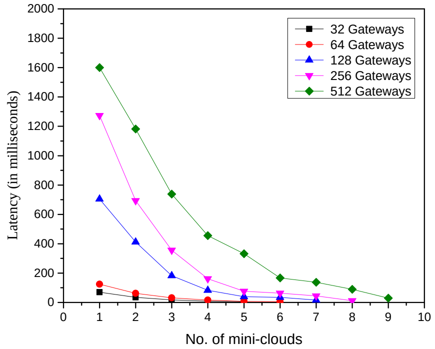

Fig. 2: Latency Vs. No. of mini-clouds

required to reduce the service latency. Figure 3 shows the overall power consumption(kW) of mini-clouds. It is observed that the power consumption is drastically changing in fog and cloud computing for data offloading. Power consumption is very less in fog tier.

VIII. CONCLUSION

In this study, we have seen the edge-centric data collection for IoT services. It was observed that the service latency and power consumption in fog computing environment are significantly lower than the cloud computing environment for a large number of real-time, low latency applications. In the future, we plan to extend this work by promising other algorithms to select centroids and compare those algorithms in a fog computing environment.

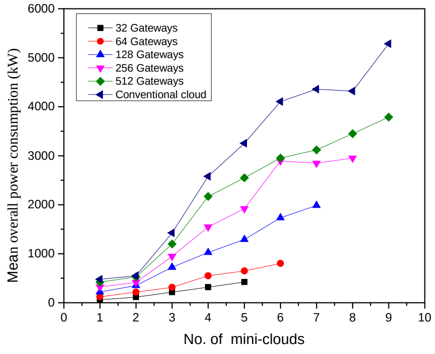

Fig. 3: Overall power consumption Vs. No. of mini-clouds

ACKNOWLEDGMENT

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

REFERENCES

[1] K. Ashton,“That internet of things thing in the real world, things matter more than ideas,”RFID Journal, 2009.
[2] H. Vestberg, “CEO to Shareholders: 50 Billion Connections, ”2010.
[3] D. Evans, “The Internet of Things: How the Next Evolution of the Internet is Changing Everything ,”, White Paper, 2011.
[4] A. Sinaeepourfard ,J. Garcia , X. Masip-Bruin , E. Marn-Tordera , J. Cirera , G. Grau , and F. Casaus, “Estimating Smart City sensors data generation,”, Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Vilanova i la Geltru, pp. 1-8 , 2016.
[5] A. Zanella , N. Bui , A. Castellani , L. Vangelista ,and M. Zorzi,“Internet of Things for Smart Cities ”, vol.1, pp. 22-32, 2014.
[6] F. Bonomi ,R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things,”Procedding of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16, 2012.
[7] Q. Duan, Y. Yan, and A. V. Vasilakos, A survey on service-oriented network virtualization toward convergence of networking and cloud computing, IEEE Trans. on Network and Service Management, vol. 9,no. 4, pp. 373-392, December 2012.
[8] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, An SMDP-Based Service Model for Interdomain Resource Allocation in Mobile Cloud Networks, IEEE Trans. on Vehicular Technology, vol. 61, no. 5, pp. 2222-2232, Jun 2012.
[9] T. H. Noor, Q. Z. Sheng, A. H. H. Ngu, and S. Dustdar, Analysis of web-scale cloud services, IEEE Internet Computing, vol. 18, no. 4, pp. 55-61, July 2014.
[10] A. V. Dastjerdi and R. Buyya, Compatibility-aware cloud service composition under fuzzy preferences of users, IEEE Trans. on Cloud Computing, vol. 2, no. 1, pp. 113, Jan 2014.
[11] J. Xiao, H. Wen, B. Wu, X. Jiang, P.-H. Ho, and L. Zhang, Joint design on dcn placement and survivable cloud service provision over all-optical mesh networks, IEEE Trans. on Communications, vol. 62, no. 1, pp. 235-245, January 2014.
[12] 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.

[13] 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.

[14] 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, 2014.

[15] S. Sarkar; S. Chatterjee; S. Misra, ”Assessment of the Suitability of Fog Computing in the Context of Internet of Things,” in IEEE Transactions on Cloud Computing , 2015.

[16] S. Sarkar, and S. Misra , ”Theoretical modelling of fog computing: a green computing paradigm to support IoT applications,” in IET Networks, vol. 5, pp. 23-29, 2016.