QoS-aware Service Provisioning in Heterogeneous Fog computing supporting IoT Applications

Prasenjit Maiti
Department of Computer Science & Engineering
National Institute of Technology, Rourkela
Odisha, India
pmaiti1287@gmail.com

Hemant Kumar Apat
Department of Computer Science & Engineering
National Institute of Technology, Rourkela
Odisha, India
hemant.rimt@gmail.com

Mohammed Altowayti
Department of Computer Science & Engineering
National Institute of Technology, Rourkela
Odisha, India
mohammed.altowayti@gmail.com

Sampa Sahoo
Department of Computer Science & Engineering
National Institute of Technology, Rourkela
Odisha, India
sampaa2004@gmail.com

Bibhudatta Sahoo
Department of Computer Science & Engineering
National Institute of Technology, Rourkela
Odisha, India
bibhudatta.sahoo@gmail.com

Ashok Kumar Turuk
Department of Computer Science & Engineering
National Institute of Technology, Rourkel
Odisha, India
akturuk@gmail.com

Abstract—Internet of Things (IoT) applications is several interconnected services. Fog Computing provides a distributed platform to data processing and resource provisioning for IoT services. Ensuring Quality of Services (QoS) in terms of energy and low-latency for these services is essential. Deploying IoT services in the Fog computing infrastructure is a difficult process due to heterogeneous Fog nodes. In this paper, we introduced the service provisioning problem in heterogeneous fog computing environment to meet the QoS constraint and meta-heuristic approach is proposed to solve it efficiently.

Index Terms—Fog Computing, Service Provisioning, Quality of Service, Internet of Things, Resource Allocation

I. Introduction

Fog computing [1] is a concept of a system that receives the services on their way going to cloud, check if they are needed to be processed and stored in the cloud then forward them, otherwise process them in the fog itself and response with a feedback to the smart devices. It is argued that fog system reduces the bandwidth required to service the smart devices by not sending every request directly to the cloud but rather process the services that are possible to be serviced in the fog devices i.e. routers. By this way, not only bandwidth, but also many other system parameters are conserved, and it is claimed that this system guaranties an optimal quality of service. Lowering the cost and raising the efficiency is also claimed to be achieved through this system of computing which results in better and more efficient user experience [2]. Fog computing is also concerned whenever the concept of real-time response servicing is required. We take for example, a self-driving car in which route information calculation has to be done in every part of a second (real-time response requirement) to decide when to increase/ decrease the speed, when to take a turn and which direction, when to stop and how to follow traffic signals, and many other parameters that require instant response [3]. In this case, if we depend in the cloud, knowing that the delay of time is huge for the data to travel to cloud, get processed, and travel back to the system for feedback, there are very high chances of accidents which can result in life loses. Fog computing gives a solution to this problem and many similar problems like airplane navigation systems, hospital equipment, and so on. How fog computing solves that is by processing the data in the edge devices themselves without the need for sending them to the cloud, and as edge devices are close to the smart devices to the limit that they are wirelessly connected, the time delay is enormously less compared to cloud computing approach. Fog computing has a strong relationship with internet of things as there are huge amount of devices of daily-life usage that generate enormous size of data and require real-time response like mobile-sensors, personal computer requirements, smart homes sensors, and so on. Our work is to devise plans on which the fog and cloud work together to give best results for data processing and sending response by processing the most urgent and secretive data on the fogs, and send the rest of the data to be processed in the cloud.

II. Related Work

For the last couple of years, service placement problem is a major research challenge in both industry and academia. The problem addresses the number of services required and their placement with considering different performance metrics. Karima et al. [4] proposed service placement architecture to reduce the latency using smart service placement system that facilitates the location of services in the proper position according to specific needs. Service Orchestrator was discussed as well as some details of implementation using ILP. Yousefpour et al.[5] introduced the dynamic fog service provisioning problem which dynamically deploy the services on fog nodes or releasing the services that are already deployed on fog nodes in order to meet the expected QoS constraints. INLP is used to formulate the problem and heuristic solutions are used for simulation. Quality-of-Service (QoS) provisioning is an essential feature of next-generation networks. Skarlat et al.[6] present a conceptual fog computing framework and then model the service placement problem for IoT applications over fog resources as a optimization problem. Genetic Algorithm is used for attaining the optimal placement of services in order to minimize the communication delay and better utilization of fog resources. The optimization method produces a service placement plan which is more effective in utilizing the fog landscape resources, leading to lower execution cost when compared to the average service placement plan produced by the genetic algorithm (with the cost constituting only 40% of the cost of service placement plans produced by the genetic algorithm). The genetic algorithm produces solutions which on average experience a lower deployment delay by exploiting more cloud resources (on average 36% of the services have been run in the cloud). Quang Tran et al. [7] proposed an approach to optimize the service placement on fog landscape. A multilayer fog computing architecture is presented that optimize service decentralization on fog landscape and also improved the latency, energy consumption and network load if we compared with traditional cloud computing. The approaches are not only cost effective but also improve the utilization of virtual resources. Mahmud et al. [8] proposed a Quality of Experience (QoE) -aware application placement policy that prioritizes different application placement request according to users' expectation and calculate the capacities of fog instances considering their current status. It also facilitates placement of applications to suitable fog instances so that users QoE is maximized in respect of utility access, resource consumption and service delivery. Brito et al. [9] proposed an architecture called as fog orchestration architecture to deliver the services to the end users. Fog orchestration efficiently handles the infrastructure management node selection and service placement in fog computing paradigm. In this paper, they present architecture for service orchestration based on the core requirements of Fog Computing. Based on a virtualized environment, where Fog Nodes are capable of running virtualized and containerized applications and services, offering them access to attached/connected devices, over different communication technologies, to accomplish their task. Jose et al.[10] tried to improve the performance metrics by adopting fog computing in place of fog computing. Allocation of services to appropriate nodes is an important criterion while evaluating the performance of global file system so a multiple optimization objective has been resolved such as low latency and energy efficiency has been taken into consideration for the evaluation of performance.

III. Proposed Fog Computing Framework

As fog computing is a part of data operating between users and machines, it plays an important job in making data exchange faster and more efficient. Existing models represent different aspects of cloud/fog computing, but in this model, we concentrate on efficiency, energy consumption, and time delay as they are considered the main parameters for real-time user-devices interaction.

Tier-1 :- Smart Sensor Nodes (SSN): These nodes are responsible for gathering data from the surrounding environment like temperature, humidity, light, sound, touch, and so on. As the parameters are understood by these nodes, they transform the analogue parameters i.e. sound waves into digital signals that could be understood by the system that sensors are connected to. The basic function of these devices is to capture the analogue parameters that should be sensed, transform information into digital data, and finally sends this data to the system (fog colonies) for further processing. As the fog colonies receive data from different outgoing sources from the smart systems i.e. routers and gateways, these coordinators control the flow of data from different sources to avoid randomization, duplication, and collision of data.

Tier-2:- Fog nodes consist of embedded devices or systems for managing and processing data received by the fog colony. These systems receive the data and coordinate it till the end of servicing process i.e. listener unit which stays alarmed till any type of request reaches to the fog colony and then it notifies the needed for processing units that some request is needed to be served, watchdog unit keeps checking on all other units in case any of them went down then the reasoned unit should be informed to take a proper action, compute unit is also responsible for most of the calculations occurring in the fog systems, and database unit keeps a register of actions occurring in the fog and could store small chunks of data if they are needed in near future computation. Comes to the picture one of the most important components of the fog node which is fog action control which controls all the actions and operations in the fog colony and reasons that with the service registry, which contains data about all the

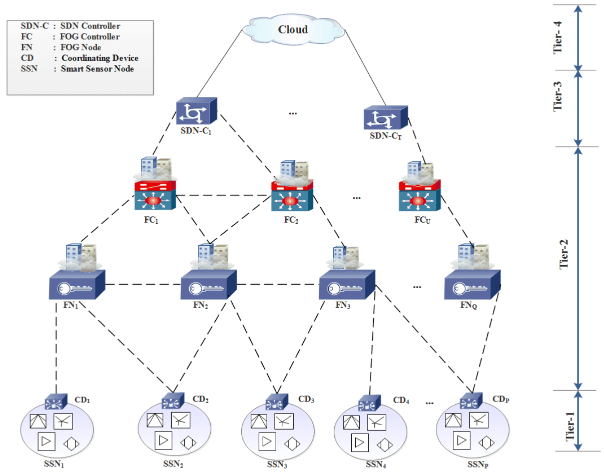

Fig. 1. Fog Computing Architecture for IoT Services

requests being services, for further computations.

Tier-3:- Fog Controllers (FC) control the functionality of all the connected fog nodes and assign each node to a specific number of operations. It plays an important role in case that if service requests are being processed in some of the fog nodes and the other nodes are idle so it assign some of the requests from the current functioning nodes to the idles nodes for load balancing. Software-Defined-Network Controllers (SDN-C is responsible for the flexibility of networks connected to it. In SDN-C layer, traffic of services is being shaped from the console that is centralized in nature without the need to be in touch with the routers or switches in the specified network.

Tier-4:- It is huge storing and processing system that is constructed through collective construction of storage spaces collected from different places through WAN or internet and supported with primary system parameters like processing units and random access memories. The most efficient functionality of the cloud is for storing data as the shared memory can be enormous, but it doesn't serve well in the functioning part not because cloud doesn't have enough computational resources but as the response time is high due to the long distance that services have to travel to reach to the cloud.

IV. Problem Statement

Service request originate from multiple users over the internet. The challenge of allocating service requests to a set of Virtual Machines (VMs) running on different fog nodes while achieving the terms and conditions stated in the Service Level Agreements (SLAs) and without degrading the Quality of Services (QoS) is referred to as the Fog Service Placement Problem (FSPP). This FSPP is a non deterministic polynomial time hard (NP-hard) problem. The virtualization technique allows one to create virtual instances of a fog node or resource, where the framework partitions the resource into multiple executions environment virtually in the form of VMs.

A. Assumptions

(i) Each VM is capable of running different number of heterogeneous services. (ii) The resource capabilities of VMs are heterogeneous. (iii) The system has adequate resources (VMs) for processing all accepted services. (iv) A Virtual Machine Monitor (VMM) is running on the top of each fog node. (v) A service is allowed to execute only on a single VM. (vi) Each VM is not capable of executing all types of services. Service Migration has not been taken into consideration.

B. Mathematical Model

A heterogeneous fog node that consists of a set of $F_N = \{F_{N_1}, F_{N_2}, ..., F_{N_n}\}$, n independent heterogeneous, uniquely addressable computing entity. It has a set of $V_M = \{V_{M_1}, V_{M_2}, ..., V_{M_k}\}$, k heterogeneous VMs, and consider each fog node has multiple VM. $S_E = \{S_{E_1}, S_{E_2}, ..., S_{E_m}\}$, m number of heterogeneous services, where each service S_{E_i} has a service length L_i in terms of millions of instructions (MI). ECT_{ij} is the expected time to compute service S_{E_i} on VM V_{M_j}. Here ψ_{ij} is a binary decision variable which is decided a service S_{E_i} is allocated to a VM V_{M_j} or not.

$$\psi_{ij} = \begin{cases} 1\ ,\ if\ S_{E_i}\ is\ assigned\ to\ V_{M_j} \\ 0\ ,\ Otherwise \end{cases}$$

Makespan is the maximum execution time among all the VMs to process all services assign to a Fog computing system. The makespan is defined as:

$$M\left(m,k\right) = \max\left(\sum_{j=1}^{k}\sum_{i=1}^{m}\psi_{ij}ECT_{ij}\right) \tag{1}$$

The energy consumption in fog nodes depends on the allocation techniques of services to a set of VMs. The total energy consumption of the Fog computing network is defined as:

$$\varphi = \sum_{j=1}^{k} E\left(V_{M_j}\right) \tag{2}$$

The objective is to minimize the makespan and energy consumption is as follows:

$$\phi = \min\left(M\left(m,k\right), \varphi\right) \tag{3}$$

V. Proposed Algorithms

The service provisioning algorithm which will select the appropriate fog nodes to process services in the fog computing framework. Here, we present First In First Serve (FIFS) and Particle Swarm Optimization (PSO) to solve FSPP. Algorithm 1 describes the selection of appropriate fog node to deploy IoT services on the basis of service arrival time. Algorithm 2 proposed hereinafter address the PSO based service placement to optimize the makespan and energy.

Algorithm 1: $FSPP\_FIFS$

Input: S_E, V_M

Output: Mapping of Service $S_{E_i} \in S_E$ to $V_{M_j} \in V_M$

1 Initialize services.

2 First service S_{E_1} is assigned to Fog controller's service queue and add services up to size of the queue.

3 Assign Service's S_{E_i} to V_{M_j} according to the service's arrival time to the queue and capability of V_{M_j} to S_{E_i} process .

4 Repeat until all the services are placed.

Algorithm 2: $FSPP\_PSO$

Input: S_E, V_M

Output: Mapping of Service $S_{E_i} \in S_E$ to $V_{M_j} \in V_M$

1 A population of service allocation vector which are randomly generated, local velocity vector (L_v) and global velocity vector (G_V), and velocity vector (V) for each particle in a population.

2 Convert all continues vector to discrete vector including the service allocation vector.

3 Calculate the fitness value$(M(m,k),\varphi)$ for each particle.

4 Update the particle's best position (P_{best}) for all particles.

5 The global best position (G_{best}) is the minimum fitness value.

6 for each particle update the particle's velocity and position

7 $V_i = V_i + L_v \times Rnd(0,1) \times (P_{best} - X_i) + G_v \times Rnd(0,1) \times (G_{best} - X_i)$

8 $X_i = X_i + V_i$

9 end for

10 $it = it + 1$

11 until $it > MAX\_ITERATIONS$

VI. Experimental Results

In this section we study the behavior of the algorithms. The simulation has been carried out in MATLAB R2017b on Intel(R) Core(TM) i7-7700 HQ CPU, 2.81 GHz processor, 64 bit operating system, and 12GB RAM with parameters set as following. The MATLAB Simulator has been widely used to evaluate various advanced proposed techniques in the literature [11], [12]. It is observed from Fig. 2. that the energy consumption in PSO is less than FIFS for FSPP. From the above results, we conclude that among the two algorithms, PSO gives a better makespan for different scenarios. A comparative summary is shown in Fig. 3, Fig. 4, Fig. 5, and it can be observed that the optimization parameters for the three cases is less in case of the PSO algorithm.

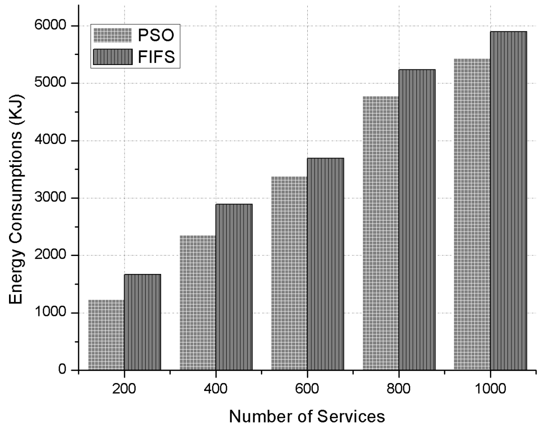

Fig. 2. No. of Services Vs Energy Consumption

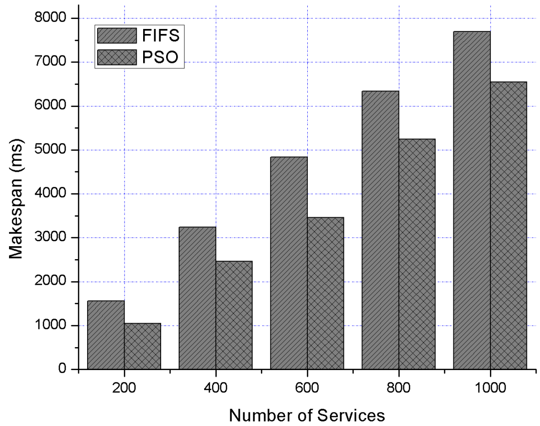

Fig. 3. No. of Services Vs Makespan

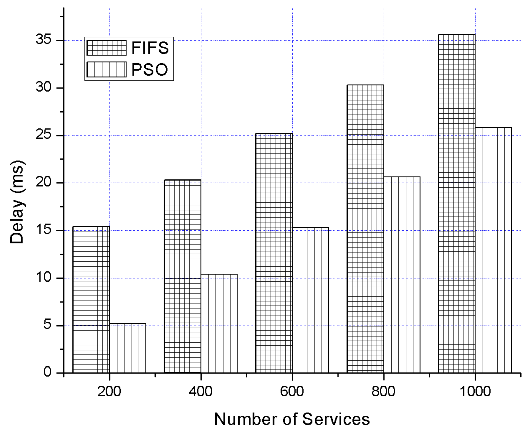

Fig. 4. No. of Services Vs Delay

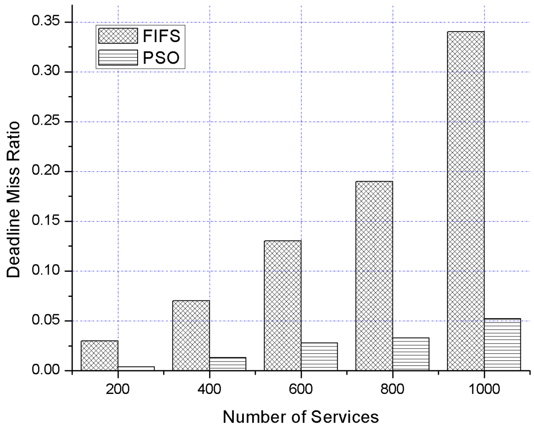

Fig. 5. No. of Services Vs Deadline Miss Ratio

VII. Conclusion

Optimization of service placement in fog computing environment considers the system parameters CPU, RAM, and storage, it also considers the processing parameters energy consumption, time delay, response time, and task status of being services, still in queue, or deadline reached to its limit. To conclude, for the processes that are urgent and damage will be caused if they get delayed, then the best approach to be considered is the PSO method, and if there is no priority and the safety of the fog colonies is the goal then FIFS is to be used.

Acknowledgment

This research was supported by Media Lab Asia (Visvesvaraya Ph.D. Scheme for Electronics and IT, Project Code-CSVSE) under the department of MeitY government of India and carried out at Cloud Computing Research Laboratory, Department of CSE, National Institute of Technology, Rourkela, India.

References

[1] F. Bonomi, R. Milito, P. Natarajan,and J. Zhu, “Fog computing: A platform for internet of things and analytics,”. In Big data and internet of things: A roadmap for smart environments , Springer, Cham, pp.169-186, 2014.

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

[3] E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwälder, “ Incremental deployment and migration of geo-distributed situation awareness applications in the fog,” In Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems , ACM, pp. 258-269, 2016.

[4] K. Velasquez, D. Perez Abreu, M. Curado, E. Monteiro, “Service placement for latency reduction in the internet of things,” Annals of Telecommunications, pp. 105-115, 2017.

[5] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, Weisheng Xie, and J. P. Jue, ”QoS-aware Dynamic Fog Service Provisioning,” arXiv preprint arXiv:1802.00800 , 2018.

[6] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, ”Optimized IoT service placement in the fog”, Service Oriented Computing and Applications, Vol 11, no. 4, pp. 427-443, 2017.

[7] Q. T. Minh, D. T. Nguyen, A. Van Le, H. D. Nguyen and A. Truong, ”Toward service placement on Fog computing landscape,” 4th NAFOSTED Conference on Information and Computer Science, Hanoi, pp. 291-296, 2017.

[8] M. Redowan, S. Narayana Srirama, K. Ramamohanarao, and R. Buyya, ”Quality of Experience (QoE)-aware placement of applications in Fog computing environments,” Journal of Parallel and Distributed Computing , 2018.

[9] M. S. de Brito et al., ”A service orchestration architecture for Fog-enabled infrastructures,” Second International Conference on Fog and Mobile Edge Computing (FMEC), Valencia, 2017, pp. 127-132, 2017.

[10] J. Santos, T. Wauters, B. Volckaert and F. De Turck, ”Resource provisioning for IoT application services in smart cities,” 13th International Conference on Network and Service Management (CNSM), Tokyo, pp. 1-9, 2017.

[11] D. A. Chekired and L. Khoukhi, ”Smart Grid Solution for Charging and Discharging Services Based on Cloud Computing Scheduling,” in IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3312-3321, Dec. 2017.

[12] [12] R. Deng, R. Lu, C. Lai and T. H. Luan, ”Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing,” IEEE International Conference on Communications (ICC), London, 2015, pp. 3909-3914, 2015.