International Journals of Advanced Research in Computer Science and Software Engineering
ISSN: 2277-128X (Volume-9, Issue-9)
Research Article **September 2019**

Efficient Task Distributed Algorithm using Virtual Machines for Fog Computing

P. Krishna Madhuri
M. Tech,
Department of CSE,
Shri Vishnu Engineering College for Women (A),
Vishnupur, Bhimavaram,
West Godavari District, Andhra Pradesh.

A. Seenu
M.Tech(PhD), Associate Professor,
Department of CSE,
Shri Vishnu Engineering College for Women (A),
Vishnupur, Bhimavaram,
West Godavari District, Andhra Pradesh.

Abstract— Fog computing is a new computing architecture, composed of a set of near-user edge devices called fog nodes, which collaborate together in order to perform computational services such as running applications, storing a signicant amount of data, and transmitting messages. Fog computing extends cloud computing by deploying digital resources at the premise of mobile devices and smart objects. In this new paradigm, management and operating functions, such as job scheduling aim at providing high-performance, cost-effective services requested by mobile devices and smart objects and executed by fog nodes. I propose a new bio-inspired optimization approach called Task Distribution Algorithm aimed at addressing the job scheduling problem in the fog computing environment. Our proposed approach is based on the optimized distribution of a set of tasks among all the fog computing nodes. The objective is to find an optimal tradeoff between CPU execution time and allocated memory required by fog computing services established by mobile devices and smart objects.

Index— Fog Computing , Task Distribution Algorithm, Mobile Devices,

1. Introduction

Fog computing is based on the idea of enabling computing directly at the edge of the cloud infrastructure rather than being transmitted to the core of the cloud (i.e. data centers) (The Network 2016). Fog computing is defined as a distributed computing infrastructure that extends computational services offered by the cloud computing to the edge of the network, as presented. Considered as a complement infrastructure to the cloud, fog computing facilitates task processing, networking and data storage between cloud data centers and mobile users.

Specifically, several applications and computing services that do not fit well with the cloud, could be performed by the fog like applications requiring reduced and predictable latency, geographically distributed applications (e.g., wireless sensor networks), intelligent transportation applications (e.g., smart connected vehicles, smart traffic lights). To serve a fog computing user, an application could be executed both by cloud components such as smart gateways, routers and data center, as well as edge components of the cloud network called fog nodes. Fog nodes are often resource-constrained devices such as base stations, access points, routers, set-top-boxes that support computational and storage resources, transmission protocols, mobility, diverse types of interface.

Fog nodes are used to meet the requirements of computational applications, which request reduced communication overheads, minimized latency, dense and large-scale distribution. One example of a fog device, IOT products manufactured by Cisco, helps developers to develop IoT applications for control systems, data aggregation, and cybersecurity. These IOT applications are hosted on a guest operating system that enables compiled code to be executed at the network edge. To support fog services, new management and operating functions such as task scheduling were introduced. These new fog functions can be offered by a set of fog servers placed at the edge of networks in selected locations like offices of local government, shopping centers or service stations.

2. Proposed System

To deal with the task scheduling problem in fog computing, I propose a new bio-inspired optimization approach named Task Distribution Algorithm to find an optimal allocation for a task's tasks among the available fog resources (i.e. fog nodes) so that I can achieve a tradeoff between the CPU execution time and the allocated memory. In this way, the response latency and bearable cost (i.e. allocated memory) can satisfy mobile users' requests. I evaluate the performance of the proposed novel optimization approach based on TDA and demonstrate its efficiency by comparing its performances with other approaches.

3. SYSTEM ARCHITECTURE

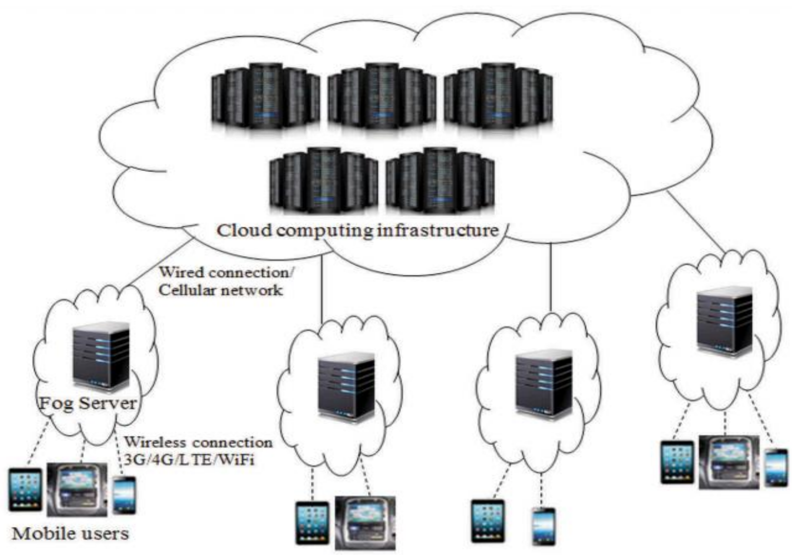

Fig 1 :- Fog computing infrastructure.

To serve a fog computing user, an application could be executed both by cloud components such as smart gateways, routers and data center, as well as edge components of the cloud network called fog nodes. Fog nodes are often resource-constrained devices such as base stations, access points, routers, set-top-boxes that support computational and storage resources, transmission protocols, mobility, diverse types of interface. Fog nodes are used to meet the requirements of computational applications, which request reduced communication overheads, minimized latency, dense and large-scale distribution One example of a fog device, IOT products manufactured by Cisco, helps developers to develop IOT applications for control systems, data aggregation, and cybersecurity. These IOT applications are hosted on a guest operating system that enables compiled code to be executed at the network edge.

4. IMPLEMENTATION

Fog Node:-

- To randomize load balancing across different fog nodes in the network .
- Load balancing is the process of scheduling tasks among a set of N physical servers
- Each server can run up to C tasks in parallel and has a waiting area of capacity B tasks.
- The improvement in the performance of a stochastic process characterized by a random decision that has to be taken.

System Model:-

- A set of N providers divided into **G** different groups
- Each managing a single fog node composed of C independent physical servers.
- $\mathbf{S^t_u}$ the state of fog node u at time t,

- The number of busy servers (i.e., on use) at time t, $\mathbf{S^t_u} \in \{0, 1, ..C\}$.
- The workload of node u at time t, $\mathbf{w^t_u}$,
- The fraction of busy servers of a given node u, $\mathbf{w^t_u} = \mathbf{S^t_u} /C$.

Task Scheduling Algorithm:-

- The basic idea of the algorithm is to schedule tasks in the task scheduling relational table with higher priority.
- Set the completion time of these tasks in the table to a larger value, and then select the fog node with the minimum completion time.
- Execute a loop from the rest of the tasks to select the task with minimum completion time to schedule
- Assign the selected task to the fog node with minimum completion time until all of the tasks are scheduled.

5. VM TASK SCHEDULING ALGORITHM

Data: Service s to be scheduled
Data: Latency constraint t requested by the service provider
Data: Users' locations $U = \{u_1, u_2, \ldots\ldots, u_m\}$
Data: Nodes $N = \{n_1, n_2, \ldots\ldots, n_k\}$
Data: Nodes' location $L = \{l_1, l_2, \ldots\ldots, l_k\}$
Data: VM types $VM = \{v_{i,1}, v_{i,2}, \ldots\ldots, v_{i,q}, \forall_i \varepsilon N\}$
Data: Array Q of quality scores for each VM
$v \varepsilon VM$
Result: The vm_s scheduled for s.
1 **begin**
2 Define a set of users' groups $G = \{g_{i,} |u_j - l_j| < \varepsilon \forall u_j \varepsilon U, \forall l_i \varepsilon L\}$
3 forall the $n \varepsilon N$ do
4 forall the $g \varepsilon G$ do
5 Estimate the network path $p_{g,n}$ with the lowest latency $\tilde{t}_{g,n}$ between g and n
6 Estimate the available bandwidth $b\tilde{w}_{g;n}$ between users in g and the node n and the required bandwidth bw_g on $p_{g,n}$
7 Compute the latency score $q_{n,l,g} = LatencyScore(t, \tilde{t}_{g,n})$
8 Compute the bandwidth score $q_{n, bw, g} = Bandwidth\ Score(bw_g, b\tilde{w}_{g,n})$
9 $q_{n,l} = \Sigma^n_{i=1} |g_i|\ q_{n,l,gi} / \Sigma^n_{i=1} |g_i|$
10 $q_{n,bw} = \Sigma^n_{i=1} |g_i|\ q_{n,bw,gi} / \Sigma^n_{i=1} |g_i|$
11 forall the $v \varepsilon n$ do
12 $q_{v, res} = ComputingScore(w, \tilde{w}_v)$
13 Compute the q_v quality score $Q[q_v] = 3/ (1/q_{v, res}) + (1/q_{n,l}) + (1 /q_{n,bw})$
14 $vm_s = Scheduler (Q)$
15 return vm_s

6. RESULTS

The task scheduling problem in fog computing, i used two performance evaluation metrics in this study namely, the CPU execution time and the total amount of memory (allocated memory) needed by all tasks expected to be executed in the fog computing infrastructure. The resource utilization performance. From the above discussion, we clearly understand that our proposed approach achieves the better performance compared to other approaches.

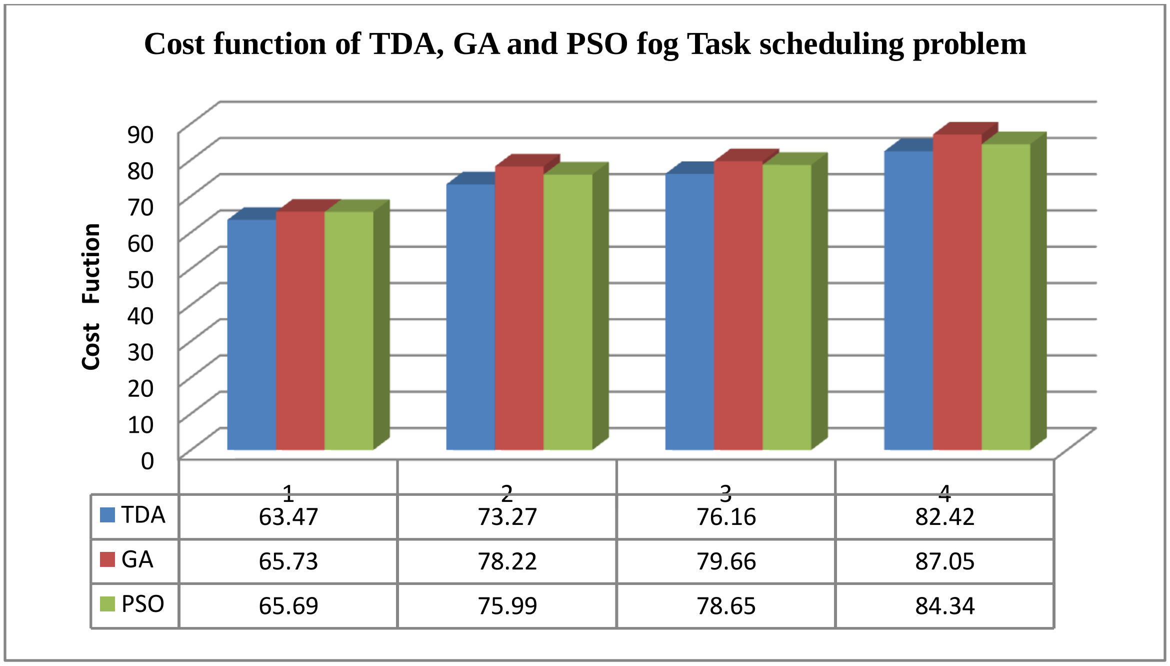

Fig: Cost Function of Task Scheduling Problem

TDA	63.47	73.27	76.16	82.42
GA	65.73	78.22	79.66	87.05
PSO	65.69	75.99	78.65	84.34

Table: Cost Function of Task Scheduling Problem

7. Conclusion

Fog computing is expected to continue to attract the attention of researchers in academia and industry given its tremendous potential in which computing resources and services are distributed in efficient fog nodes reside at the edge of the cloud computing network. In this work, I focused on the task scheduling problem in the fog computing environment to ensure the efficient execution of tasks and satisfy the service requests of mobile users. To address the task scheduling challenge for fog computing, I have proposed a new optimization method called Task Distribution. Our proposal is a population-based approach based on collaborative behaviors of the individuals of the population. To address the task scheduling problem in fog computing, I used two performance evaluation metrics in this study namely, the CPU execution time and the total amount of memory (allocated memory) needed by all tasks expected to be executed in the fog computing infrastructure. To evaluate the reliability and the efficiency of our proposed Task Distribution algorithm.

8. Future Work

I will investigate the implementation of a dynamic task scheduling approach that considers the arrival of new requests while the other requests are being executed for the fog computing environment. I also plan to investigate another dynamic aspect of task scheduling when the fog servers are mobile. Additionally, i will consider optimization of the network bandwidth because this metric could improve the overall end-to-end processing performance of the underlying fog computing infrastructure.

Bibliography

[1] A. Saeed, A. Abdelkader, M. Khan, A. Neishaboori, K.A. Harras, A. Mohamed, Argus: realistic target coverage by drones, in: IPSN, 2017, pp. 155–166.
[2] A. Essameldin, K.A. Harras, The hive: An edge-based middleware solution for resource sharing in the internet of things, in: Proceedings of the 3rd Workshop on Experiences in the Design and Implementation of Smart Objects, ACM, 2017.

[3] M. Ibrahim, M. Gruteser, K.A. Harras, M. Youssef, Over-the-air tv detection using mobile devices, in: ICCCN, IEEE, 2017, pp. 1–9.
[4] A.B. Said, A. Mohamed, T. Elfouly, K. Harras, Z.J. Wang, Multimodal deep learning approach for joint eeg-emg data compression and classification, in: Wireless Communications and Networking Conference, WCNC, 2017 IEEE, IEEE, 2017, pp. 1–6.
[5] A. Emam, A. Mtibaa, K.A. Harras, A. Mohamed, Adaptive forwarding of mhealth data in challenged networks, in: E-Health Networking, Applications and Services, Healthcom, 2017 IEEE 19th International Conference on, IEEE, 2017, pp. 1–7.
[6] 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.
[7] H. Abdelnasser, K.A. Harras, M. Youssef, Ubibreathe: A ubiquitous non-invasive WiFi-based breathing estimator, in: MobiHoc, ACM, 2015, pp. 277–286.
[8] H. Abdelnasser, M. Youssef, K.A. Harras, Wigest: A ubiquitous wifi-based gesture recognition system, in: Computer Communications, INFOCOM, 2015 IEEE Conference on, IEEE, 2015, pp. 1472–1480.
[9] H. Abdelnasser, M. Youssef, K.A. Harras, Magboard: Magnetic-based ubiquitous homomorphic off-the-shelf keyboard, in: SECON, IEEE, 2016, pp. 1–9.
[10] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for vm-based cloudlets in mobile computing, IEEE Pervas. Comput. 8 (4) (2009).
[11] K. Habak, M. Ammar, K.A. Harras, E. Zegura, Femto clouds: Leveraging mobile devices to provide cloud service at the edge, in: Cloud Computing, CLOUD, 2015 IEEE 8th International Conference on, IEEE, 2015, pp. 9–16.
[12] H. Gedawy, S. Tariq, A. Mtibaa, K.A. Harras, Cumulus: A distributed and flexible computing testbed for edge cloud computational offloading, in: Cloudification of the Internet of Things, CIoT, IEEE, 2016, pp. 1–6.
[13] A. Mtibaa, A. Emam, S. Tariq, A. Essameldin, K.A. Harras, On practical deviceto-device wireless communication: A measurement driven study, in: Wireless Communications and Mobile Computing Conference, IWCMC, 2017 13th International, IEEE, 2017, pp. 409–414.
[14] A. Mtibaa, K.A. Harras, K. Habak, M. Ammar, E.W. Zegura, Towards mobile opportunistic computing, in: Cloud Computing, CLOUD, 2015 IEEE 8th International Conference on, IEEE, 2015, pp. 1111–1114.
[15] A. Saeed, M. Ammar, K.A. Harras, E. Zegura, Vision: The case for symbiosis in the internet of things, in: Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services, ACM, 2015, pp. 23–27.
[16] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, E. Benkhelifa, The future of mobile cloud computing: integrating cloudlets and mobile edge computing, in: Telecommunications, ICT, 2016 23rd International Conference on, IEEE, 2016, pp. 1–5.
[17] A. Elgazar, K.A. Harras, M. Aazam, Towards intelligent edge storage management: Determining and Predicting Mobile File Popularity, in: Mobile Cloud, Mobile CLOUD, 2018 IEEE 6th International Conference on, IEEE, 2018.
[18] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, G.-J. Ren, Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems, IEEE Wirel. Commun. 23 (5) (2016) 120–128.
[19] C. Huang, R. Lu, K.-K.R. Choo, Vehicular fog computing: architecture, use case, and security and forensic challenges, IEEE Commun. Mag. 55 (11) (2017) 105– 111.
[20] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K.R. Choo, M. Dlodlo, From cloud to fog computing: A review and a conceptual live VM migration framework, IEEE Access 5 (2017) 8284–8300
[21] M. Aazam, E.-N. Huh, M. St-Hilaire, Towards media inter-cloud standardization–evaluating impact of cloud storage heterogeneity, J. Grid Comput. (2016) 1–19.
[22] S. Ibrahim, H. Jin, B. Cheng, H. Cao, S. Wu, L. Qi, Cloudlet: towards mapreduce implementation on virtual machines, in: Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing, ACM, 2009, pp. 65–66.
[23] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal, et al., Mobileedge computing introductory technical white project, White Project, MobileEdge Computing, MEC, Industry Initiative, 2014.