International Journal of Engineering & Technology, 7 (4) (2018) 3761-3765

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet. v7i4.12342
Research paper

Survey of resource management techniques in fog computing

Vadde Usha [1] *, Dr. Vijaya Sri Kompalli [2]

[1] *Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India – 522502*
[2] *Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India – 522502*
**Corresponding author E-mail: vaddeusha@rediffmail.com*

Abstract

The Internet of Things (IoT) generates large volume of data. Today's Cloud models offer scalability, reliability and cost effectiveness. However the large volume, variety and velocity of data that is being generated by IoT devices challenges the long delay links between cloud data center and IoT devices. Fog Computing is an extension to the cloud computing, introduced by CISCO is a distributed paradigm. The basic idea is to deploy Cloud Computing infrastructure closer to the things (sensors/smart devices) that produce and act on data. This fascinating concept brings together the latest technologies like Cloud, Edge, IoT etc. In Fog computing, resource management is an important factor for better utilization of available resources and also providing optimal service for IoT applications. This study focuses on the recent research works undertaken in the resource management area of Fog computing and also compares various edge computing paradigms. At the end, other issues that are left as future challenges are highlighted.

Keywords: *Edge Computing; Fog Computing; Internet of Things; Resource Management.*

1. Introduction

IoT facilitated interaction among the things and humans, which empowered the smart cities and smart factories. The IoT allows the objects, including electronic devices, home appliances and all types of sensors to interconnect and exchange the data. The vast amount of data collected from these heterogeneous devices needs to be processed so that further decision can be made. However, processing of vast amount of data by the IoT devices alone is difficult because of their limited computational power, storage and energy resources. Offloading in this context refers to the transferring of resource intensive computational tasks to cloud for processing. IoT devices may offload processing on to the centralized Cloud, utilizing its opulent resources.

The recent 5 years have witnessed tremendous advancements in the field of IoT. The devices that are connected in various applications like health monitoring, industrial automation, network of sensors and actuators, games, surveillance are increasing in billions. The data generated by these devices creates massive network traffic towards centralized cloud data centers creating a bottleneck. Usually the centralized Cloud data centers are placed at distant to the IoT devices. Because of massive network traffic and distance, latency increases, causing degradation of Quality of Service (QoS). Minimizing latency is important for most of the IoT applications. For example, in health monitoring application real time response is necessary so that doctors can treat the patient in time. Edge Computing permits processing and storage at the edge of the network [1]. Edge computing has been proposed to resolve the high latency and degradation of QoS issues of centralized cloud computing [2].Edge Computing makes use of edge devices (e.g. routers, base stations, and wireless access points) with storage and processing capabilities to serve applications. Edge Computing facilitates faster responses and reduces the traffic towards the core network. Hence the Edge computing based architecture can be used for cloud-IoT based applications.

Fog Computing is a Distributed and edge paradigm which brings the cloud services close to things that produce data [3]. In the Fog computing environment, the devices that provide the resources for processing are called fog nodes. Any network components like routers, switches with storage, processing & network connectivity can be a fog node. The fog node can be placed anywhere within the range of cloud to things. The proximity of fog node provides several advantages in terms of latency, power consumption, and network traffic. Hence the delay sensitive IoT applications can benefit from the fog computing. Rather than offloading every time to the centralized cloud, the smart devices can take the benefits of closer resource rich fog node [4]. Figure 1 shows fog computing paradigm.

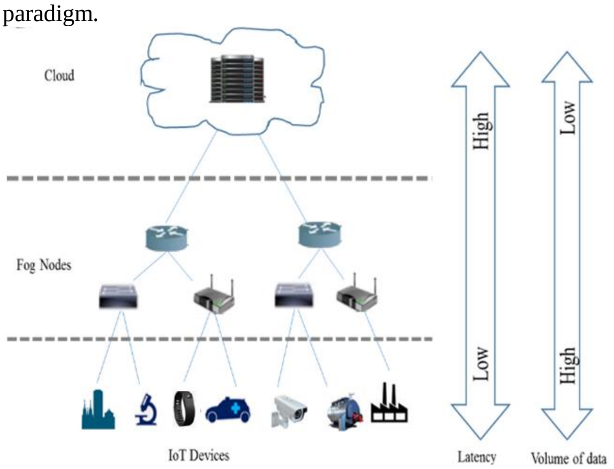

Fig. 1: Fog Computing.

The concept of Fog computing is similar to the existing edge computing paradigms except in few factors. Hence we compare these paradigms with fog computing.
The further concepts of the study include Section 2 highlights about the related computing paradigms and compare with Fog Computing. Section 3 presents recent research efforts in resource management in Fog Computing. Section 4 discusses the research challenges and Section 5 provides the final conclusion.

2. Related edge computing paradigms

Several Computing Paradigms have been proposed using the idea of Cloud& Edge Computing. The Edge layer can be implemented between the end devices and cloud in various ways [5]. Based on Edge layer implementation the paradigms can be classified as

- Mobile Edge Computing (MEC)
- Mobile Cloud Computing(MCC) using Cloudlet
- Fog Computing.

The first paradigm is Mobile Edge Computing. In MEC the servers with storage and processing capabilities are deployed in base stations of cellular network [6]. Cloud Computing services are offered by MEC, within the purview of Radio Access Network (RAN) in close proximity to mobile subscribers. Mobile subscribers can benefit with low latency and high bandwidth. MEC supports two (device and edge location) or three (device, edge and cloud) tire hierarchical framework for application deployment [7]. Within RAN, context information like user location is available for content and application developers [8].
The second paradigm is Mobile Cloud Computing (MCC). It allows computational intensive applications like language translators, image processing to run in smart or mobile devices. However, these mobile devices are resource constrained [9]. Hence, MCC offloads computationally intensive tasks to resource rich Cloud data centers for processing and storage. But MCC faces many challenges [10], including limitations of mobile devices in accessing the cloud. Latency is a major issue. Cloudlet [11] is a server with similar capabilities to data center, but on a lower scale, present in closer proximity of user for example in a coffee shop. Cloudlet supports three tire(device, edge and cloud) hierarchical framework for application deployment[7]. Therefore, most often MCC offloads to the Cloudlet to address the latency issues.
Third paradigm is Fog Computing. It makes use of Fog Computing Nodes (FCNs) at the edge of the network and near to the things that produce data. As a consequence multitier (three or more) application deployment is possible [7]. As most of the IoT applications require real time responses, physical proximity also comes into play [8]. Fog nodes can be placed closer to IoT sensors/devices compared to other edge components (Cloudlets / MEC servers). Because of this close proximity characteristic, compared to other related models, Fog computing is considered as more appropriate for IoT. Comparison of edge computing paradigms has been presented in Table 1.
To capture the full potential of Fog for real time latency sensitive applications, several challenges need to be addressed. Among these, resource provisioning while maintaining the QoS requirements of application is the most important challenge to be considered.
In this study, we describe the state-of-the-art works in resource provisioning focusing on the metrics they have considered.

3. State-of-the-art works

Olena Skarlat et al. proposed a frame work for fog resource provisioning using fog colonies [12]. The authors have used the basic structure of fog computing as described in [13]. Their work introduces fog colonies that are micro data centers, which contains fog cells and fog orchestration control node. Fog cells are software components running on fog devices and are managed by fog orchestration control node. These fog colonies are supported by a cloud fog control middleware. This middleware manages the task execution in the Cloud. Fog cells can receive the task request and allocate its resources if available or promulgate request to fog orchestration control node in turn to cloud fog control middleware. REST API is used by fog cells for data transfer and control actions. The authors have compared their proposed resource provisioning approach with baseline approach. They have used the CloudSim framework [14] to simulate the fog landscape. Time shared and space shared resource provisioning policies are possible with CloudSim. The proposed architecture along with the time and space shared resource provisioning of fog cells provided optimal utilization of the fog cells and decreased the delay up to 39% when compared to baseline approach and had shorter roundtrips and makespans. QoS constraints to the task requests have not been considered and authors specified it as future scope of work.
In other works of Olena Skarlat et al. have extended their work on fog landscape [15]. They have proposed Fog Service Placement Problem (FSPP), in which service placement will be based on proactive scenario. The objective of their work is to utilize fog landscape to the maximum extent while satisfying the application requirements. In FSPP, when fog control node receives a service request, it places the service in its fog colony. If enough resources are not available, it sends services to closest fog colony or to the cloud. While placing the services, fog control node checks that the placed services are not exceeding the given percentage of available resources. It also checks the deadlines of application, prioritizes the application with closer deadline. In their evaluation they have extended iFogSim[16] to simulate fog landscape. The Application and AppModule classes are modified so that application deadlines and deployment times can be considered. Their optimized model is implemented using IBM CIPLEX solver and Java. FSPP reduces the execution cost by 35% with respect to cloud as 70% of services utilize the fog landscape. Improving the optimization model by adding constraints about the availability, reliability and cost of resources and finding an efficient fog colony for service propagation are suggested as future scope of the work by them.
Mohammad Aazam et al. proposed MeFoRE for Resource Estimation (RE) and enhancing the QoS[17]. RE helps in controlling underutilization of resources. Their work is based on historical records of customers. RE is based on Relinquish Rate(RR).Quality of Experience(QoE) records and Network Promoter Score(NPS) are considered in enhancing the QoS. Whenever a service request is received by a fog node, it has to decide how much amount of resources has to be allocated based

Table 1: Comparison of Edge Computing Paradigms

	Fog Computing	MEC	Cloudlet
Nodes	Access points, Routers, Switches and Gateways	Servers at Base Station	Low scale data center
Location of Nodes	Anywhere between end device and cloud	Radio network controller	May run directly in device (Close proximit y to the user)
Context awareness	Medium	High	Low
Proximity (application deployment in N-tier hierarchy)	N=3 or more Multiple hops	N=2 or 3 One hop	N=3 One hop
Access Mechanism	Wireless Networks, Mobile Networks, Bluetooth, Wi-Fi	Mobile Networks	Wi-Fi or Fixed Access
Internode Communication	Yes, possible	Partial	Partial
Mobility Support	Yes	Yes	Yes

on RR, NPS and QoE. For a new customer default RR & NPS are applied. If overall NPS is greater than the default value, more priority is given to overall NPS. For a particular service, if NPS given by the customer is available then this will be given priority. CloudSim is used in implementing proposed algorithms and applied on real IoT Crawdad trace[18] and Amazon EC2 service. MeFore estimates the resources then allocates based on previous records of the customers and eventually service quality is enhanced and resource underutilization is minimized.

Hong et al. proposed a programming model called Foglets [19]. Their earlier work describes the preliminary version of foglets programming model called mobile fog [20]. They extended mobile fog to provide algorithms for the discovery and deployment of application components. Foglets provide incremental resource provisioning from sensors to the cloud. Whenever a request comes for an application component from a child node, discovery algorithm finds the list of fog computing nodes with matching capacity constraints and the list will be sent to a child node. If any fog node with the requested application component is already deployed is available, then the child node chooses this node and sends a joining message. If already deployed fog nodes are not available, then the best fog node among available is chosen for deployment. This work also provides the mechanism for application components migration, taking into account sensor mobility and application dynamic computational needs. Foglets are suitable for situation awareness applications. The Foglets are implemented using C++, Ubuntu operating system, docker container [21] and RocksDB [22]. They have used ZMQ[23] and protobuf libraries from google[24] to implement communication protocol. They have shown that discovery and deployment can be executed in 0.93s, joining in 65ms and proactive migration in 6ms. To check the scalability of migration mechanism by developing a large scale application using Foglets was suggested as their future scope of the work.

Huaqing Zang et al. proposed a three layer game frame work in the scenario of multi Data Service Operators(DSO), multi Fog nodes and multi Authorized Data Service Subscribers(ADSS)[25]. In general DSOs provide services to ADSSs. To improve QoS, Fog nodes are introduced and maintained by different infrastructure providers. However ADSSs cannot interact directly with the Fog nodes. Instead, they request the service from DSO and DSOs in turn are required to prompt the Fog nodes to serve ADSSs by paying some amount. Purchasing optimal amount of resources from Fog nodes to satisfy the QoS requirements of ADSSs is challenging for DSOs. Another issue here is Fog nodes may compete with each other for serving ADSS and ADSSs compete with each other for better services. There should be an optimal matching between resources requested by ADSS and resources offered by Fog nodes. To address these challenges the authors have adopted Game Theory approach for interaction among DSOs, Fog nodes and ADSSs. The objective of their work is to obtain optimal utility for each DSO, FN and ADSS. Stackelberg Subgame was used between DSOs and ADSSs, Contract theories Model Hazard was used between DSOs and Fog nodes and Student Project Allocation matching Subgame for achieving stable resource allocation between Fog node and ADSS. Based on the proposed game frame work, the authors have shown that with the number of ADSSs increasing the total utility is increasing irrespective of computing data size. Utility of ADSSs in Fog computing is higher than utility of ADSSs in Cloud computing.

Luiz et al. addressed the problem of resource allocation while considering geo-location and application classes [26]. They have considered hierarchical infrastructure composed of cloudlet and cloud. The authors have classified applications into two classes, namely near real time and delay tolerant. They have selected Electroencephalography Tractor Beam Game (EEGTBG) for near real time and Video Surveillance Object Tracking (VSOT) application for delay tolerant in assessing how application QoS can be effected with different scheduling strategies in the context of user mobility. Three different scheduling strategies, namely Concurrent, First Come First Serve (FCFS) and Delay Priority were used for showing how different strategies will impact applications differently, when many users are moving towards a single cloudlet. The authors evaluated these strategies using iFogSim. The results have shown that VSOT application suffered with higher delays with concurrent and delay priority strategies, while maintaining low delay with FCFS. Delay Priority strategy is effective for application in the lower delay class. The authors have specified that uncertainty in bandwidth and processing time of application modules are the critical factors in scheduling.

In Table 2, summary of the reviewed works has been given.

Table 2: Summary of Reviewed Works

Work	Objective	Problems Addressed	Methods Used	Measurements	Fog Terminology / Tools
Olenaskarlat et al.[12]	Optimal utilization of available Fog based computational resources	Minimization of delay.	System model for Fog resource provisioning.	39% delay decreased compared to baseline approach	• Fog colonies, fog cell, fog orchestration control node, cloud-fog control middleware. • CloudSim
Olenaskarlat et al.[15]	IoT services placement on Fog resources considering QoS requirements.	• Maximum utilization of Fog landscape • Time (deadline of application execution time)	Linear programming model.	• Proposed algorithm uses 70% of Fog landscape • 30% execution cost reduced when compared to cloud	• Fog colonies, Fog cell, control node, Fog computing management system • iFogSim
Mohammad aazam et al. [17]	To provide resource estimation on the basis of historical records and enhance QoS using previous quality of experience records	• Mitigation of resource underutilization • Enhancing QoS.	Resource estimation algorithm	• Underutilization of resources minimized • Quality of service improved.	CloudSim
Hong et al. [19]	Programming infrastructure for	• Resource provisioning in	• Discovery & deployment	• Discovery & deployment - 0.93	• Foglets • C++,Uubuntu

	computational continuum extending from the sensors to the cloud called foglets	steps • Mobility • Dynamic computational needs of an application	protocol • Join protocol. • api and handlers for communication &qos driven migration.	sec • Joining - 65msec	operating system, Docker container, RocksDB
Huaqing zhang et al. [25]	Frame work for resource management in multi fns, DSOs, ADSSs scenario.	Stable and optimal utilities/payoffs for each DSO, fog node and ADSS	• Stackelberg subgame • mural hazard modeling • Student project allocation matching subgame	ADSS utility in Fog is high compared to Cloud	Data Service operator, Fog node, authorized data subscriber
Luiz et al. [26]	Resource allocation considering hierarchical infrastructure, analyzing application classes along with different scheduling policies.	• Mobility • Current cloudlet load and request requirement	• Delay priority • Concurrent scheduling • FCFS • Above scheduling algorithms are compared.	Delay priority algorithm maintains lower delays for delay sensitive applications	• Cloudlet • iFogSim

Fog Computing plays an important role in serving widely distributed IoT devices/Sensors. The works carried out so far identified and addressed many important aspects of fog computing. But some more issues still exists, which have to be focused to improve the fog computing. In the next section some of the challenges in Fog computing have been discussed.

4. Research challenges

Even though multiple works have considered context information [17] in estimating the resources, few more aspects of contextual info are still to be explored. Some of them are time, mobility aspects, network traffic, and available resources

Reliability in terms of availability of efficient Fog nodes, fault tolerance, etc can be focused so that the high performance of Fog computing can be achieved [27]

Support for Mobility in Fog computing brings in many more challenges. Future works can consider service migration issues like the combination of proactive and reactive migration and using the context information like the current location of the user in service migration, so that a best Fog node can be chosen among the eligible target nodes [28].

Fog environments must have the capability to add and remove resources dynamically because of the mobile nature of processing nodes that may frequently join and leave the network. New Programming models and architectures are necessary to address the dynamic configuration [29].

In fog computing while managing the resources there may be multiple objectives like minimizing completion time and to minimize power consumption cost. But, minimizing completion time means allocating more resources in contrast to minimize power consumption. Thus, there is a tradeoff between these two objectives. Addressing contradictive objectives is a challenging issue [30].

5. Conclusion

Rapid growth of IoT based applications have triggered the need for more sophisticated ways to handle and process large volumes of data that is being generated by the IoT devices. Cloud-IoT solution has addressed this by providing its rich resources for processing and storage. But this solution suffers from longer delays. However, quick response is expected by the many of the IoT applications. Hence we need a mechanism to pull Cloud resources closer to IoT devices to avoid delay. Fog Computing address this by placing the resources at the edge of the network. In this study, we have discussed the need for Fog computing and also compared Fog computing with the related edge paradigms. Then a survey of state-of-the-art research efforts in Fog resource management has been presented and summarized. Some of the research challenges which still persist and require attention are also discussed.

References

[1] Varghese, B., N. Wang, S. Barbhuiya, P. Kilpatrick, and D.S. Nikolopoulos, “Challenges and opportunities in edge computing”, in *Proceedings of the IEEE International Conference on Smart Cloud*, 20–26,2016.

[2] P Garcia Lopez, A Montresor, D Epema, A Datta, T Higashino, A Iamnitchi, M Barcellos , P Felber , E Riviere, “Edge-centric Computing: Vision and Challenges”, ACM SIGCOMM Computer Communication Review, vol. 45 n. 5, pp. 37-42,2015 https://doi.org/10.1145/2831347.2831354.

[3] Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are Cisco white paper 2015.

[4] S. Sarkar, S. Chatterjee, S. Misra, "Assessment of the suitability of fog computing in the context of internet of things", IEEE Transactions on Cloud Computing, vol. PP, no. 99, pp. 1-1, 2015.

[5] K Dolui, S K Datta,Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing” in Global Internet of Things Summit (GIoTS), 2017.

[6] ETSI PORTAL, “Mobile-Edge Computing – Introductory Technical White Paper” 2014.

[7] Guenter I. Klas, Fog Computing and Mobile Edge Cloud Gain Momentum Open Fog Consortium, ETSI MEC and Cloudlets, 2015

[8] A Ahmed, E Ahmed, “A survey on mobile edge computing” in 10th international conference on Intelligent Systems and Control (ISCO), IEEE, 2016.

[9] Z. Sanaei, S. Abolfazli, A. Gani, R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges", Communications Surveys & Tutorials IEEE, vol. 16, pp. 369-392, 2014. https://doi.org/10.1109/SURV.2013.050113.00090.

[10] H Qi, A Gani, “Research on Mobile Cloud Computing: Review, Trends and perspectives”, in second international conference on Digital Information and Communication Technology and its applications, 2012.

[11] M Satyanarayanan, P Bahl, R Caceres, “The Case for VM-Based Cloudlets in Mobile Computing Computing”, Pervasive Computing, IEEE, vol. 8, no:4, pp. 14-23,2009 https://doi.org/10.1109/MPRV.2009.82.

[12] O. Skarlat, S. Schulte, M. Borkowski, P. Leitner, "Resource Provisioning for IoT Services in the Fog", ninth IEEE International Conference on Service Oriented Computing and Applications (SOCA 2016), pp. 32-39, 2016. https://doi.org/10.1109/SOCA.2016.10.

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

[14] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software - Practice and Experience, vol. 41, pp. 23-50, 2011. https://doi.org/10.1002/spe.995.

[15] O Skarlat, S Schulte, M Borkowski," Towards QoS-Aware Fog Service Placement", in 1st international conference on Fog and Edge Computing (ICFEC), IEEE, 2017

[16] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things Edge and Fog Computing Environments", Cloud Computing and Distributed Systems Laboratory The University of Melbourne Tech. Rep. CLOUDS-TR-2016–2, 2016.

[17] M Aazam, M St-Hilaire, CH Lung,I Lambadaris, "MeFoRE: QoE based resource estimation at Fog to enhance QoS in IoT" in 23rd international conference on Telecommunications (ICT), IEEE, 2016

[18] F. Songwei, Z. Yan, CRAWDAD dataset due/packet-delivery (v. 2015-04-01) traceset: packet-metadata, April 2015.downloaded from http://crawdad.org/due/packet-delivery/20150401/packet-metadata

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

[20] Kirak Hong , David Lillethun , Umakishore Ramachandran ,Beate Ottenwalder , Boris Koldehofe, Mobile fog: a programming model for large-scale applications on the internet of things, Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing, pp.15-20, 2013 https://doi.org/10.1145/2491266.2491270.

[21] S. Hykes. What is docker? https://www.docker.com/whatisdocker/. Accessed: 2015-09-30.

[22] Rocksdb. http://rocksdb.org/. Accessed: 2016-01-12.

[23] P. Hintjens. ZeroMQ: Messaging for Many Applications. "O'Reilly Media, Inc.", 2013.

[24] protobuf: Google's data interchange format. https://github.com/google/protobuf. Accessed: 2016-01-02.

[25] H Zhang, Y Zhang, Y Gu, D Niyato, Z Han, "A Hierarchical Game Framework for Resource Management in Fog Computing" in IEEE Communications Magazine, Vol 55, no 8, 2017 https://doi.org/10.1109/MCOM.2017.1600896.

[26] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, M. Parashar, "Mobility-aware application scheduling in fog computing", IEEE Cloud Computing., vol. 4, pp. 26-35, Mar./Apr. 2017. https://doi.org/10.1109/MCC.2017.27.

[27] Mahmud R, Kotagiri R, Buyya R, "Fog Computing: A Taxonomy, Survey and Future Directions" In: Di Martino B., Li KC., Yang L., Esposito A. (eds) Internet of Everything. Pp. 103-130,Internet of Things

[28] C Puliafito, E Mingozzi, G Anastasi, "Fog Computing for the Internet of Mobile Things: Issues and Challenges" , IEEE International Conference on Smart Computing (SMARTCOMP), 2017 https://doi.org/10.1109/SMARTCOMP.2017.7947010.

[29] A. V. Dastjerdi, R. Buyya, "Fog computing: Helping the Internet of Things realize its potential", IEEE Comput., vol. 49, no. 8, pp. 112-116, Aug. 2016. https://doi.org/10.1109/MC.2016.245.

[30] C Mouradian , D Naboulsi, S Yangui, R H. Glitho, M J. Morrow, P A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-art and Research Challenges" in IEEE Communications Surveys & Tutorials ,Vol: PP,no.99, 2017.