Recent Advancement of Auto-Scaling in LTE M2M Communication.

Sunita T. N, Bharathi Malakreddy A

Abstract*: *Lately Machine to Machine (M2M) Communication has gathered huge research interest because of its peculiar nature of communication without any or less human intervention. With the increase in wide variety of devices and application, there is huge change in traffic patterns of Machine Type Communication (MTC) system. Existing traditional Long-Term Evolution (LTE) network will not be able to handle these growing demands of the bandwidth and network availability. There are some challenges in the existing network like latency, scalability, reliability, interference and delay, which degrade the Quality of Service (QoS). Hence to address these issues would require some advanced network resource management capabilities such as Network Functions Virtualization (NFV), Software Defined Networking (SDN). These technologies would help the operators to provide efficient services to consumer. In this literature we present survey of auto-scaling the resources required for LTE communication using SDN, NFV and Machine Learning (ML) for facilitating MTC, along with its requirements, existing work and challenges. This paper first describes in brief about SDN/NFV and its limitations. Then review the existing work and their applicability to MTC along with open problems and finally some future research in this area.

Keywords: Auto-Scaling, Machine to Machine, SDN, NFV, Machine Learning.

I. INTRODUCTION

As the industry is evolving rapidly, so do the internet service requirement is also increasing. It’s not like one kind of service can satisfy various industries such as healthcare, automotive, manufacturing etc. all these industries require a wide range of network service as compared to the existing LTE network service provider. The traditional LTE network is not able to satisfy the various requirement of industry is due to some existing challenges such as interference, delay, latency, scalability and reliability. To efficiently address the service requirement of the industry, the traditional LTE network system must undergo some enhancement. Recently there were various solutions emerged in the industry like NFV and SDN which are based on virtualization technology. Long-Term Evolution (LTE) network functions such as Packet Data Network Gateway (PGW), Serving Gateway (SGW), Mobility Management Entity (MME) and different various functions which were part of hardware solution, can now be moved into clouds using virtualization technology [1],[20].

SDN intends to isolate network management functions from network forwarding functions. SDN gives you a global picture of networking and its infrastructure. Like the required infrastructure, the kind of services and applications being delivered, the network policies and rules which are required for delivery. SDN enforces programmable network control to provide efficient orchestration and automation of network services.

When you look at the traditional network, you will see a variety of hardware elements like Routers, Switches, Packet core nodes and Access nodes. Considering the increase of usage of mobile devices and increased demand for network resources. Whenever a new network services needs to be launched, it is becoming difficult to accommodate new hardware element, integrate, deploy and allocate power. And this in turn led to capital investment challenge and skills required to integrate and operate complex hardware-based applications has become a rarity. And the additional burden of hardware expiry, which would lead to procure and deploy new hardware in order to replace the existing ones without any revenue benefit.

To address these issues NFV was introduced. NFV is network architecture concept that utilizes virtualization technology to replace the network functions which are provided by hardware with Virtual Machines or Virtualized Network Functions (VNF), where in different network functions are implemented in one or more virtual machines, which can be combined to create networking communication services. This solution not only gives you an opportunity to solve some of the Telco-Industry issues, but also there are challenges to meet the Quality of Service (QoS) and reliability of the LTE system.

Keeping up the quality of service with low cost is a big challenge to the industry because of variety of service requests and different types of users. Generally, day time would require high availability of the service as compared to night time where the network load is low. And sometimes there might be sudden bursts in the traffic in the middle of nowhere. So, to adjust the resource availability along with such workload is not an easy task. Auto-Scaling technique is more suitable for these kinds of requirements [1].

Revised Manuscript Received on December 13, 2019.

***** Correspondence Author

Sunita T.N*, Department of Computer Science, RGIT, Visvevaraya Technological University, Bangalore, India. Email: sunita.neelagiri@gmail.com

Bharathi Malakreddy A, Department of Computer Science, BMSIT&M, Visvevaraya Technological University, Bangalore, India. Email: bharathi_m@bmsit.in

Retrieval Number: B10301292S19/2019©BEIESP
DOI: 10.35940/ijitee.B1030.1292S19

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In this literature a survey of some of the past and undergoing research on the auto scaling of LTE M2M communication is presented and discuss some of the challenges in auto-scaling, load balancing and research solution proposed to enhance the efficiency of the MTC using load balancer. Identify some of the open challenges and future research topics.

This paper is organized as; Section II provides insight of challenges in the autoscaling technology. Section III discusses about autoscaling in cloud environment. Section III discusses about the M2M based load balancing solutions and challenges. Section V discusses some of the future research directions and finally in the section VI conclusion.

II. CHALLENGES IN AUTO-SCALING

Auto-scaling the network resources is also not an easy task to meet the expected QoS and maintain low Service Level Agreement (SLA) violations. There are different kinds of challenges:

A. Challenge in Reliability

With the demand for high service availability and low SLA violations. Sometimes there can be delay in scaling time, which leads for violation of SLA [2]. Sometimes the scaling decision is made within the stipulated time, but Virtual Machine (VM) start-up time may take longer than usual, which would result in SLA violation. Hence there is a room for improvement to meet the demand of dynamic changes of service availability.

B. Challenge in Stability

To achieve the demand for system reliability, the VMs are scaled in and out frequently, which could result in increase in system overloads. This can be resolved by maintaining the capacity limits or scaling timeouts to reduce the unnecessary scaling events [1].

C. Challenge in Energy Saving

A recent survey shows that resource usage of the data centres is around 15% to 30% only, which is nothing, but maximum physical resources are unutilized or wasted [3]. Resource and Service providers have been facing numerous challenges in provisioning and service availability.

Penchung Tang et al [1] proposed an approach where the scaling decision are based on reinforcement learning methodology. In this work author converted the workload of VNFs as a discrete time services and the auto scaling decision was based on Markov Decision Process (MDP) model. This reinforcement learning approach lacked the usage of historic traffic data pattern and scaling decisions.

III. AUTOCALING CLOUD ENVIRONMENT

Autoscaling is the process, where the required amount of resources for the application is automatically handled depending on the usage ability of the application. Also, to avoid the resource utilization is neither underutilized or over utilized. And maintain high level of SLA. [20].

Autoscaling cannot be defined by certain steps that if you follow them then autoscaling could be accomplished. Autoscaling depends on various factors like, what type of application you are dealing with, what is the level of SLA agreement, techniques followed to achieve autoscaling, which cloud service model is used by the consumer, expected attributes (Scalability, availability, reliability etc).

Fig 1. Shows a high-level view of autoscaling in cloud computing, where we have tried to categories autoscaling into 5 buckets. [20].

A. Level of Autoscaling in Cloud

Depending on the SLA and quality of attributes for which the consumer has agreed upon. The autoscaling would be applied on different cloud service models.

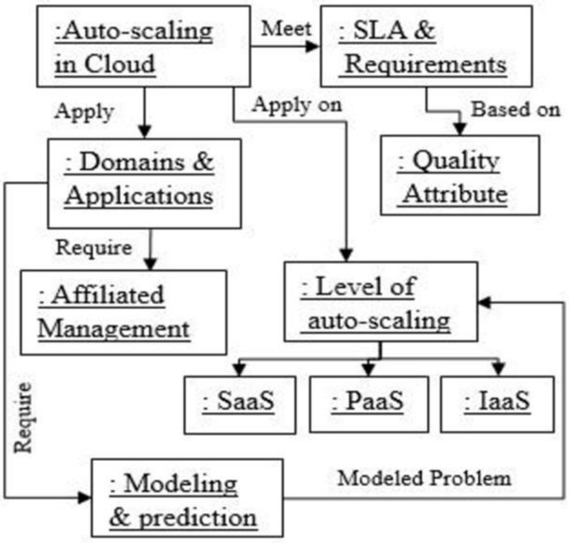

Fig. 1. **High level view of Autoscaling in cloud computing.**

- *Infrastructure-as-a-Service (IaaS)*: Here the computing infrastructure is provided to consumer by the provider over the internet. Consumer has the control on deploying an application, required storage and computing power, operating system and network. Also, can run different software. Autoscaling is applied depending on the vendor specific needs. Example. Amazon EC2.
- *Software-as-a-Service (SaaS):* Here the consumer doesn't need to worry about installation and running the application. Everything is taken care by the service provider. This kind of model is also referred as on demand software. Autoscaling is completely managed by cloud. Example. Google Apps.
- *Platform-as-a-Service (PaaS):* Here the consumer has very little control, consumer is provided with computing platforms such as operating system, programming language, database etc. Cloud manages most of the things. Autoscaling is provided by cloud which can be used by the consumer. Example. AWS Elastic Beanstalk.

B. Quality Attributes

Quality attributes play a decoding factor in building an autoscaling mechanism. The attributes like resource availability, security, performance all these attributes can make a difference in autoscaling in cloud. Example video on demand, which require maximum bandwidth and performance is a key factor for auto-scaling mechanism.

C. Domains and Applications

There are variety of application. Resource allocation depends on the type of application. Some application might need resource on demand only. Streaming applications might require high bandwidth and high availability of the resource. Hence, depending on various applications required availability of the resource, autoscaling measures need to be taken.

D. Affiliated Management

It is related to cloud computing mechanism, which deals with how resources are governed, and various optimization techniques used to reduce the operation cost, monitoring techniques used for autoscaling and security measures used in autoscaling of cloud resources.

E. Modelling and Prediction

The techniques that help in better management of cloud computing. There are so many modeling and prediction techniques available to increase the performance of the application and which are used as autoscaling features in cloud computing.

In the cloud environment different applications will be hosted and each application may have its own resource requirement. There are situations where the applications might use the resources for two hours per day then for the remaining 22 hours the resources will not be utilized. Also, there is a possibility that the application requirement might be more than the one VM, in such case the performance will be degraded. To address these kind of issues *Mohan Murthy M K et al [4]* proposed threshold-based autoscaling. In this work the author considered CPU and RAM utilization as the key parameters to increase or decrease the resources for the application. Predefined max and min CPU and RAM utilization threshold value is declared and accordingly whenever the resource utilization crosses these threshold values, either the RAM or CPU capacity will be increased or decreased.

Main drawback is that threshold values are fixed and predefined. It is not necessary that all the applications must cross these threshold values. For a more optimal results the threshold values should be dynamic, depending on either the history or feedback of the resource utilization.

Research and studies show that auto scaling in telco clouds have also been explored. *Tuan Phung-Duc et al [5]* proposed a heuristic based algorithm to dynamically scale the VNFs based on the number of jobs in the system. In cellular network along with the VNFs, traditional network equipment's are also deployed, hence the author considers even legacy equipment capacity in deciding the auto-scaling of VNFs. As this method is reactive in nature, will not be optimal solution for dynamic traffic patterns.

Various network functions such as firewall, switches, router, etc. are mostly hardware-based functionalities. And if there is demand for additional network resources then new hardware's need to be procured, integrated and deployed. This makes the traditional networks to be complex and time consuming. Hence VNF concept was introduced to replace the costly vendor hardware.

Now that VNF has been implemented and depending on the load network functions can be scaled up and down respectively. Also, auto-scaling in cloud computing has been introduced, which will automatically scale the required amount of resources depending on the load. And auto-scaling can reduce lot of operational cost.

The traditional auto-scaling mechanism follows a threshold-based approach. But threshold approach is not suitable for dynamic traffic pattern. More suitable for predictable traffics pattern. Hence the author *Sabidur Rahman et al [6]* proposed Machine Learning (ML) based classifier which would proactively take scaling decisions ahead of time, instead of traffic predictions.

The author is trying to address the traditional VNF auto scaling issue by a supervised machine learning classifier. Here the classifier is trained based on the historical VNF scaling decisions with respect to measured network loads. The outcome of this VNF classifier is proactively come up with the number of VNF instances required to address the incoming traffic loads. Also, the author showcased the performances of different virtualization technologies used for auto-scaling i.e. virtualization hypervisor (Xen, KVM) versus container virtualization (Docker, LXC).

Main drawback was that operational and leasing cost analysis was done assuming a same pattern of traffic loads, which could be extended to dynamic traffic as well. Also, the paper explored only horizontal auto scaling of VNFs. Exploring vertical auto-scaling still needs to be explored.

IV. RESULTS AND DISCUSSIONS

Machine type Communication (MTC) is the communication between the devices, refrigerators, computers, mobile phones and other sensor devices without any human intervention. In M2M network massive devices have been deployed, basically these devices will be availing various services like smart home, meter reading and so on through internet. These kind of M2M services would require different bandwidth and also response time varies. M2M networks comes with challenges like congestion due to bursty traffic, meeting the QoS and so on [7].

To resolve these kinds of issues, load balancing technique have been implemented in M2M networks, one such technique was CPU based load balancer which would create a new VM based on predefined workload threshold. This kind of technique led to service bottleneck and Single Point of Failure (SPOF). This was suitable for dynamic and bursty traffic pattern.

A. Load Balancing

The advantage of having load balancer is that, the workload is distributed between the nodes and thereby reduce the service response time. Depending on the workload the load balancer will either kick start the server to satisfy the incoming requests or turn off the servers to maintain the resource efficiency. The two categories that come under load balancing technique are reactive and proactive.

- *Reactive Methods:* Here the name itself says to react on some variations. Whether the server nodes should be brought up or down is decided based on resource utilization and a given threshold [8]. The author proposed a load balancing scheme which considered features such as time varying and over utilized resources and depending on the resource usage would dynamically assign some weights. And depending on these weights, decision was made to bring the server up or down. The author [9] proposed an adaptive load method which considered different threshold parameters such as sensor data generation rating and network conditions for distributing the load among the server nodes. The reactive method was not suitable for dynamic traffic load patterns as there was always some delay in responding to load.

- *Proactive Methods:* In proactive load balancing methods, the decision of load balancing is done based on prediction of resource utilization [10]. The author *Yu, Lei et al* [11] considered stochastic characteristics for predicting resource demand and proposed a stochastic load balancer. Prediction based load balancing schemes sometimes proved to be inefficient because of inaccurate predictions for bursty traffic patterns.

B. *SDN based load balancer*

Ideal way of handling the network overload and congestion is to isolate the control plane and data plane. And SDN provides this flexibility and can be programmatically updated as well. SDN controller is a one stop central place where you get the complete picture of the network infrastructure like network topology, link usage and so on. Using which the flow table is reconfigured and defined in each switch.

Table- I: Related work and contribution

References	Contribution	SDN	NFV	Cloud	ML	Short Comings	Results
Penchung Tang et al [1]	Proposed an approach where the scaling decision are based on reinforcement learning methodology	•	•	•		Approach lacked the usage of historic traffic data pattern and scaling decisions	The proposed algorithm outperforms as compared to static threshold and voting policy in SLA guarantee of the network service and resource efficiency.
Mohan Murthy M K et al [4]	Proposed threshold-based auto scaling			•		Threshold values are fixed and predefined	Results show that the analytical model provides a quick way to help mobile operators to plan and design network optimization strategies without wide deployment, saving on cost and time.
Tuan Phung-Duc et al [5]	This work utilizes heuristic algorithm to dynamically scale the VNFs based on the number of jobs in the system		•	•		As this method is reactive in nature, will not be optimal solution for dynamic traffic patterns.	The results show that network operators can estimate the budget for desired system performance and optimize the strategy of deployment and there by saving cost and time.
Sabidur Rahman et al [6]	Proposed Machine Learning (ML) based classifier which would take proactively scaling decisions	•	•	•	•	That operational and leasing cost analysis was done assuming a same pattern of traffic loads, which could be extended to dynamic traffic as well	Explored practical SD-WAN use-case with a backbone network showing that the proposed method yields lower leasing cost for network leasers compared to prior works

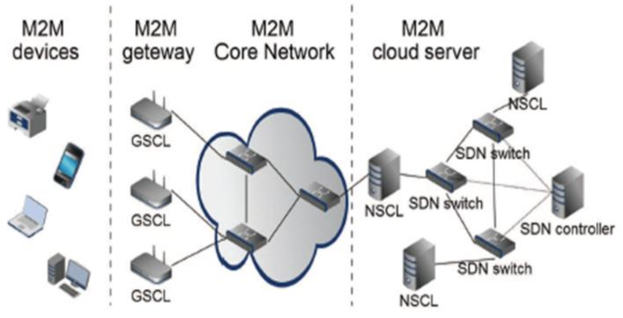

Fig. 2. **SDN enabled M2M Networks**

C. Load balancing issues

The Fig.2 shows SDN based M2M network. The MTC requests coming from MTC devices (MTCD) to cloud servers can be divided into three phases. In the initial phase the request is sent to the Gateway Service Capability Layer (GSCL) [12]. Based on the type of service required GSCL would either forward the request directly or collect number of requests and process these requests as batch process. In the second phase the GSCL would send these requests to cloud server through Network Service Capability Layer (NSCL) to manage MTC requests. In the last phase NSCL would either itself process the request or would initiate a different NSCL in datacenter to process the request, based on the load NSCL could be deployed or removed in the datacenter. The conventional SDN based load balancer faced bottleneck and Single point of failure issues. The next concern would be how to maintain the load balance of these cloud servers. After the introduction of SDN in to M2M architecture for load balancing, the SDN has efficiently distributed the client requests across the servers [13]. SDNs are the best suited for load balancing because SDN controller provides a centralized global picture of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

the network infrastructure. Also, the controllers can be updated and reconfigured easily. With SDN you can programmatically change network policies of rules for rerouting the traffic and scaling

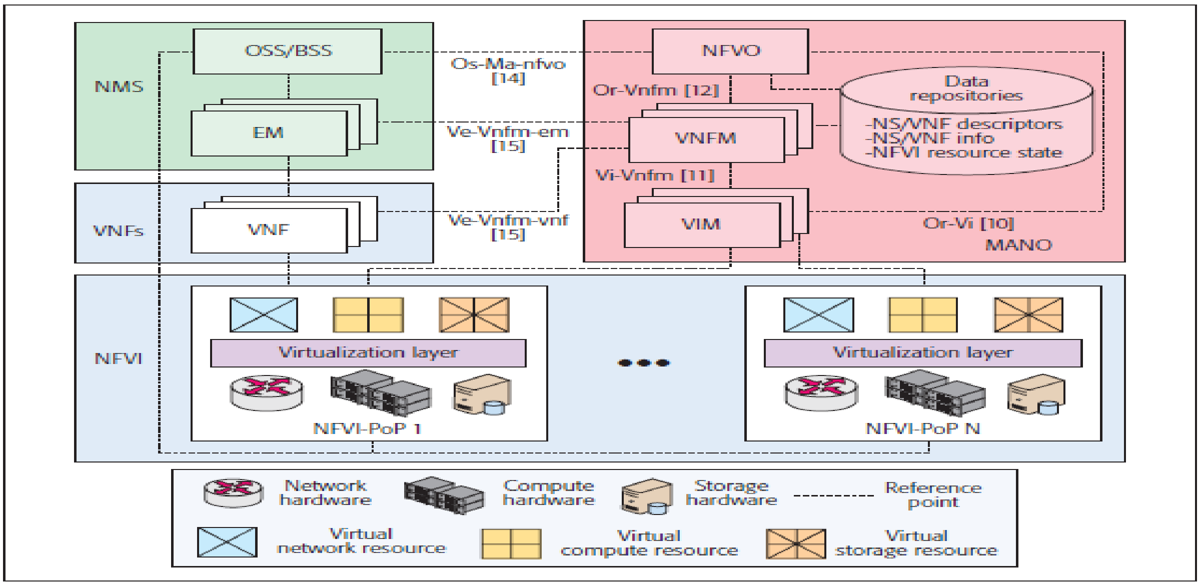

Fig. 3. **Architectural diagram of ETSI NFV framework.**

Table- II: Related work and contribution

References	Contribution	SDN	NFV	Cloud	ML	Short Comings	Results
Yu-Jia chen et al [7]	Proposed traffic aware load balancing scheme for M2M networks	•		•		For more optimal results the threshold values could be decided on the history or feedback of the resource utilization.	Results show that the analytical model provides a quick way to help mobile operators to plan and design network optimization strategies without wide deployment, saving on cost and time.
Yu, Lei et al [11]	Considered stochastic characteristics for predicting resource demand and proposed a stochastic load balancer.			•		Prediction based load balancing schemes sometimes proved to be inefficient because of inaccurate predictions for bursty traffic patterns	Proposed scheme can reduce service response time up to 50% compared to the existing load balancing scheme.
Adamuz-Hinojosa et al [19]	Proposed various options and procedures for scaling NS in NFV framework		•	•		Too many computation and information exchanges are required to attain a instantiation level to trigger the autoscaling	The options for automatically scaling a NS with the NFV framework are limited by the way the NSD is constructed. During its lifecycle, an NS instance only can move among the instantiation levels defined in the NSD, so their design is critical to ensure an effective automated scaling. Hence the author gave a brief idea on designing a effective NSD.
Duc-Hung Luong et al [18]	Tried to address the issues caused by reactive autoscaling, by predictive autoscaling for telco industries		•	•		Prediction based load balancing schemes sometimes proved to be inefficient because of inaccurate predictions for bursty traffic patterns	With prediction based autoscaling, by prediction-based analysis for the required resource the scaling decision is done before itself, hence you never see 100% resource usage for allocated resource.

up and down of VMs. Now depending on the load there was frequent processing of SDN controller for updating the flow tables and which resulted in overhead of the system [14]. To resolve these issues author came up with wild card rule timeout, which would actually delete the flow rules after a fixed timeout. This technique did not consider the variety of M2M service requirements and link loading [15].

Then the Yu-Jia chen et al [7] proposed traffic aware load balancing scheme for M2M networks, which actually uses the underlying SDNs instant traffic identification feature and reroute the traffic accordingly. And whenever the performance exceeds the threshold the policy flow tables are updated.

Fig. 3 shows the architectural view of the ETSI NFV framework used for providing Network Services (NS). Framework is mainly divided into 3 domains. Network Management Systems (NMS), Management and Orchestration (MANO), Virtualized Network Functions (VNF), and Network Function Virtualization Infrastructure (NFVI).[19]

NFVI is nothing but the group of various resources that form the cloud. VNFs are the one which run on top of NFVI. Virtualization layer provides abstraction between the virtual resources and VNFs. MANO is mainly concerned with management of virtual resource deployment and operation tasks. NMS is mainly concerned with non-virtual resource management.

Adamuz-Hinojosa et al [19] discussed on various options and procedures for scaling NS in NFV framework. The main drawback of the procedure is too many computation and information exchanges are required to attain an instantiation level to trigger the autoscaling. *Duc-Hung Luong et al* [18] tried to address the issues caused by reactive autoscaling, by predictive autoscaling for telco industries. Provisioning the required services before the actual demand of the network. The autoscaling was based on relative metrics of both IT resources and telecom resources. Sometimes had to adjust resources to match required services. Hence a little bit of horizontal and vertical scaling research is needed in order to exactly match the resource availability. *Naghmeh Dezhabad et al* [17] literature focuses on firewall services in cloud. The author tried to improve the scaling ability of the firewall services in cloud, with the learning-based reinforcement and genetic algorithm for autoscaling the firewall services. This approach lacked to consider the different traffic patterns.

V. OBSERVATIONS

There have been lot of research to develop auto-scaling to support MTC in LTE. But there are still some of open issues which needs to be resolved to achieve optimal auto-scaling for MTC without any issues. This section gives some insight and direction where research needs to be done.

A. Narrow band IoT

NB-IoT was mainly designed to support those MTCDs which have low mobility, low power, low throughput, low coverage and delay-tolerant applications. There has been some research to improve the battery life, reduce the delay, increase the throughput. But there are still some open issues. As the bandwidth for NB-IoT is low and hence coverage would be low. Still there is an opportunity to increase to coverage by using the existing architecture. As the bandwidth is low, sometimes the transmitted data might get lost in between. There is an opportunity to make sure whatever transaction that happen using the existing resource are precisely completed. There is still opportunity to use historical data for auto-scaling in NB-IoT. [16].

B. Software Defined Networking (SDN) based M2M

SDN are basically used in wired network and help in reducing the overhead by abstracting the underlying infrastructure from the data plane. There have been efforts to reduce the delay in accessing the resources which does not involve the complexities of software and hardware. MTC applications have their own priorities, requirement and services, along with different traffic characteristics. SDN controllers would need this kind of information for allocating the resources more efficiently. But the problem is how do you provide such huge data information to SDN controllers.

VI. CONCLUSION

In this survey we investigated issues caused by MTC on scaling into cloud environment. Did a brief review on challenges in auto-scaling, provided a high-level view of auto-scaling in cloud computing. Did a comprehensive survey on MTC scaling into virtual machines using various technologies like SDN/NFV. Also did a survey on some of the recent existing work and improvements to support auto-scaling MTC into VMs.

The outcome of this survey is that, the better way to handle the autoscaling of MTC into VMs is to learn and recognize the historic scaling decisions and traffic patterns and then apply some machine learning concepts to make a proactive decision in autoscaling.

REFERENCES

1. Tang, Pengcheng, Fei Li, Wei Zhou, Weihua Hu, and Li Yang. (2015) "Efficient auto-scaling approach in the telco cloud using self-learning algorithm." In 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1-6. IEEE.
2. Greene, Wedge, and Barbara Lancaster (2007) "Carrier-grade: Five nines, the myth and the reality." Pipeline magazine.
3. Delimitrou, Christina, and Christos Kozyrakis.(2014) "Quasar: resource-efficient and QoS-aware cluster management." In ACM SIGARCH Computer Architecture News, vol. 42, no. 1, pp. 127-144. ACM.
4. Murthy, MK Mohan, H. A. Sanjay, and Jumnal Anand. (2014) "Threshold based auto scaling of virtual machines in cloud environment." In IFIP International Conference on Network and Parallel Computing, pp. 247-256. Springer, Berlin, Heidelberg.
5. Phung-Duc, Tuan, Yi Ren, Jyh-Cheng Chen, and Zheng-Wei Yu. (2016) "Design and analysis of deadline and budget constrained autoscaling (DBCA) algorithm for 5G mobile networks." In 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 94-101. IEEE.
6. Rahman, Sabidur, Tanjila Ahmed, Minh Huynh, Massimo Tornatore, and Biswanath Mukherjee. (2018) "Auto-Scaling Network Resources using Machine Learning to Improve QoS and Reduce Cost." arXiv preprint arXiv:1808.02975.
7. Chen, Yu-Jia, Li-Chun Wang, Meng-Chieh Chen, Pin-Man Huang, and Pei-Jung Chung. (2018) "SDN-enabled traffic-aware load balancing for M2M networks." IEEE Internet of Things Journal5, no. 3: 1797-1806.
8. Chen, Liuhua, Haiying Shen, and Karan Sapra. (2014) "RIAL: Resource Intensity Aware Load balancing in clouds." In IEEE INFOCOM 2014-IEEE Conference on Computer Communications, pp. 1294-1302. IEEE.
9. Miao, Ye, Serdar Vural, Zhili Sun, and Ning Wang. (2016)"A unified solution for gateway and in-network traffic load balancing in multihop data collection scenarios." IEEE Systems Journal 10, no. 3: 1251-1262.
10. Nae, Vlad, Radu Prodan, and Thomas Fahringer. (2010) "Cost-efficient hosting and load balancing of massively

multiplayer online games." In 2010 11th IEEE/ACM International Conference on Grid Computing, pp. 9-16. IEEE.

11. Yu, Lei, Liuhua Chen, Zhipeng Cai, Haiying Shen, Yi Liang, and Yi Pan. (2016) "Stochastic load balancing for virtual resource management in datacenters." IEEE Transactions on Cloud Computing.
12. Corici, Marius, Hakan Coskun, Asma Elmangoush, Agus Kurniawan, Tong Mao, Thomas Magedanz, and Sebastian Wahle. (2012) "OpenMTC: Prototyping Machine Type communication in carrier grade operator networks." In 2012 IEEE Globecom Workshops, pp. 1735-1740. IEEE.
13. Corici, Andreea Ancuta, Ranjan Shrestha, Giuseppe Carella, Asma Elmangoush, Ronald Steinke, and Thomas Magedanz. (2015) "A solution for provisioning reliable M2M infrastructures using SDN and device management." In 2015 3rd International Conference on Information and Communication Technology (ICoICT), pp. 81-86. IEEE.
14. Bradai, Abbas, Kamal Singh, Toufik Ahmed, and Tinku Rasheed. (2015) "Cellular software defined networking: a framework." IEEE communications magazine 53, no. 6: 36-43.
15. Lin, Tu-Liang, Chen-Hao Kuo, Hong-Yi Chang, Wan-Kun Chang, and Yi-Ying Lin. (2016) "A parameterized wildcard method based on SDN for server load balancing." In 2016 International Conference on Networking and Network Applications (NaNA), pp. 383-386. IEEE.
16. Al-Kaseem, Bilal R., and Hamed S. Al-Raweshidyhamed. (2017) "SD-NFV as an energy efficient approach for M2M networks using cloud-based 6LoWPAN testbed." IEEE Internet of Things Journal 4, no. 5: 1787-1797.
17. Dezhabad, Naghmeh, and Saeed Sharifian. (2018) "Learning-based dynamic scalable load-balanced firewall as a service in network function-virtualized cloud computing environments." The Journal of Supercomputing 74, no. 7: 3329-3358.
18. Luong, Duc-Hung, Huu-Trung Thieu, Abdelkader Outtagarts, and Yacine Ghamri-Doudane.(2018) "Predictive Autoscaling Orchestration for Cloud-native Telecom Microservices." In 2018 IEEE 5G World Forum (5GWF), pp. 153-158. IEEE.
19. Adamuz-Hinojosa, Oscar, Jose Ordonez-Lucena, Pablo Ameigeiras, Juan J. Ramos-Munoz, Diego Lopez, and Jesus Folgueira. (2018) "Automated network service scaling in nfv: Concepts, mechanisms and scaling workflow." IEEE Communications Magazine 56, no. 7: 162-169.
20. Ordonez-Lucena, Jose, Pablo Ameigeiras, Diego Lopez, Juan J. Ramos-Munoz, Javier Lorca, and Jesus Folgueira. (2017) "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges." IEEE Communications Magazine 55, no. 5: 80-87.

AUTHORS PROFILE

Sunita T N, received her B.E degree and M.Tech degree in Computer Science & Engineering from Visvesvaraya Technological University (VTU), Belgaum, Karnataka. She is a Research Scholar in the Dept.of CSE at BMSIT& M, Bengaluru under Visvesvaraya Technological University. Currently, she is working as Assistant Professor in the Dept.of CSE, Rajiv Gandhi Institute of Technology College, Bengaluru, Karnataka. Her areas of interest include Mobile communication, IOT, Networking, Machine-To-Machine Communication, Data Structures and Machine Learning.

Bharathi M A, has B.E degree in Computer Science and Engineering, M.Tech in CSE and Ph.D. in Computer Science and Engineering, University of JNTU Hyderabad. Presently working as Professor Dept.of CSE at BMSIT&M, Bengaluru. Worked as visiting faculty for IGNOU and Infosys campus connect program. Areas of interest are Wireless Sensor Network, Medical Imaging, IoT, Big Data and Cloud Computing. Published papers in IEEE, Springer, Elsevier and LNSE.

Retrieval Number: B10301292S19/2019©BEIESP
DOI: 10.35940/ijitee.B1030.1292S19

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication