Performance and resource-efficiency Evaluation Framework for IoT Edge Computing

Muneeb, Erkki, Tanesh, Mika*,
*Centre for Wireless Communication, University of Oulu, Finland
{firstname.lastname}@oulu.fi

***Abstract*—Internet of Things (IoT), together with fifth-generation (5G) mobile systems**

***Index Terms*—Security; Edge Computing; Blockchain; IIoT; Industry 4.0.**

I. Introduction

[Overall introduction to the topic, starting from very general observations of current trends related to IoT, Cloud and Edge computing, and discussion on what kind of different performance and efficiency requirements are originating from different use cases of IoT. Focus on how important it is to be able to evaluate performance and efficiency of different architectural choices (centralized, MEC, local-edge).]

Internet of Things (IoT) is gaining popularity day by day according to its new models introduced in the wireless communication where smart objects such as sensors , actuators , mobile phones which interact with each other using unique addressing schemes to attain common goals[1].

The increase in the growth of low latency IoT applications such as online gaming , real-time data analysing, etc. is causing existing traditional centralised cloud computing architecture to face many problems. The main reason is the high end-to-end latency between the end devices and distant centralised cloud servers [2]. In addition , IoT devices generate around us exchange huge amount of privacy sensitive , safety critical data which is collected and saved in the distant data centres are appealing targets of various attacks[3].Finally , after facing these issue in the traditional cloud, the end user need to keep its data more private and less vulnerable[4][5].

Therefore , advancement towards the edge computing (EC) is better than centralized cloud to address the concern of end user latency , the use of computational resources, data privacy , and security at the edge of the network[6].

II. Related Work

[In this section, write about state-of-the-art situation of IoT, Cloud Computing, Edge Computing and Fog computing. Define different system models (traditional, mec, "local edge"). Introduce also the most relevant simulation tools.]

Related Work : IoT devices such as sensors, actuators , wearable devices causing huge amount of data processing , storage and communication at the edge network. According to the CISCO, around 50 billion devices will be introduced by 2020 in the internet framework[7][8]. Therefore, IoT sensors produce huge amount of data which is offloaded to the centralized clouds for processing , analyzing .On the other hand , offloading billions of end devices data to and from cloud is a challenging issue in IoT-centralised cloud model[9]. Another limitation in IoT-cloud paradigm is the real time interaction which is due to the geographically long distance between end-to-end devices and cloud cause delay in communication [10] for latency sensitive IoT applications. Currently, researchers are continuously investigating the process of using edge capabilities to support IoT needs in a better way. Fog computing is different from edge computing and provides tools for distributing, orchestrating, managing, and securing resources and services across networks and between devices that reside at the edge. Edge architecture places servers, applications, and small clouds at the edge. Fog jointly works with the cloud, while edge is defined by the exclusion of cloud [11]. Cisco proposed Fog computing framework which pushes the centralized cloud services close to the edge devices which generates the data such as sensors and actuators. Fog computing support the latency sensitive applications. Therefore, Fog computing adds extra layer between end devices and centralized cloud and also provide security and privacy for private data such as healthcare, vehicle communication , user location information[7][11]. Fog nodes consist of network devices which perform computational tasks and data storage capabilities in the same way as centralized cloud.

Aazam at el . Illustrate the concept of fog cloud computing and its architecture for low latency IoT applications. He also compared the performance metrics such as processing delay, processing cost , processing capability and task length using cloud and fog computing architecture. Researcher used Cloudsim toolkit and Boston University representative internet topology generator (BRITE) network topology [12]. Puthal at el. Proposes a novel load balancing technique to authenticate the EDCs and find less loaded EDCs for task allocation. The proposed load balancing technique is more efficient than other existing approaches in finding less loaded EDCs for task allocation. The proposed approach not only improves efficiency of load balancing; it also strengthens the security by authenticating the destination EDCs [13]. Khakimov at el. proposes edge computing network model structure for fog applications to measure the workload of the network nodes in terms of latency, distribution computing power etc[14]. Banomi et al. Proposed fog computing paradigm which supports the future IoT latency sensitive applications. Author also highlighted the relationship between fog computing and cloud computing in the environment of IoT[7]. Lera at ael.

Proposes a simulator called YAFS for different IoT architecture in Fog computing. Researcher also, compare the fog nodes performance evaluation in terms of latency, cloud and edge policies, network infrastructure with other available fog computing simulators[11].

Edge computing is referred to local layer framework. Edge computing has similar features like fog computing, but edge computing is more focused on the end devices. Edge computing enhance the cloud services efficiently such as storage , computing , processing management close to one hop away from IoT devices in the local network such as such as the WiFi access points or gateways [15]. Open Edge computing defines EC provides computations services at the local network through small data centre close to the local IoT network. Some challenges like privacy , latency and connectivity that EC can handle efficiently .As the EC is one hop close to the local users , latency in edge computing is lower than cloud computing . The availability of the services for the local IoT devices are higher than cloud as users do not need to wait to get a service from a traditional cloud [15].

Such as next generation cars require real-time processing to make quick decision as it will generate 1GB data per second .Many other use cases processing requirements e.g Airplanes , Video browsing would be a challenging factor to handle such amount of data processing in traditional cloud computing and provide low latency as well[16]. Therefore, edge computing approach is required in these use cases to fulfill the latency and network congestion solution.

A. Models

1) IoT Cloud Model: The IoT and traditional cloud topology is shown in Fig 1. Traditional IoT-cloud architecture consist of three layers. Local layer contains low power IoT devices, Core layer contains router and switches to route the data to the server-cloud layer where the IoT devices data is offloaded, analysed, processed, computed and stored. Industrial IoT applications such as automation, transportation , robotics require decentralized systems to support location awareness , scability , ultra-low latency latency communication(URLLC), geo-distribution. Some latency sensitive IoT use case require less than 1ms and high reliability [17]. Therefore, cloud-IoT based architecture faces various limitations and vulnerabilities

2) MEC Model : Figure 2. illustrates the MEC architecture. MEC pushes the edge services such as storage , computing near local devices such as smart phone . The MEC architecture is proposed by (European Telecommunication Standard Institute) ETSI for future 5G networks to offload the low power IoT and mobile devices data from traditional cloud based architecture to the edge computing. The purpose of MEC is to support the cloud computing services into the existing base stations in order to enable for latency sensitive application over mobile network. MEC research focus on Radio Access Network (RAN) based architecture. Also , the collaboration between MEC and IoT architecture reduces the physical infrastructure and virtual distance , support scalability and security as compared to traditional cloud computing architecture[18]. Also , the deployment of MEC hosts at serves located within

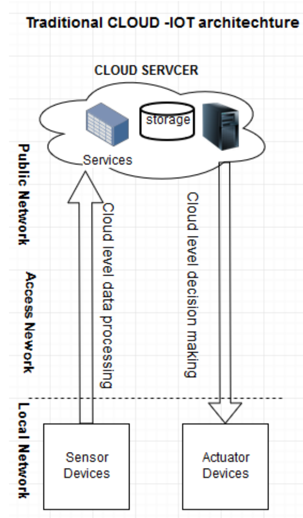

Fig. 1: Traditional Cloud-Iot Model .

or near the access network base station also has its limitations in the current model. On the other hand , Pushing data closer to the edge requires a distributed network for thousands of smart vehicles etc which is a challenging factor in MEC installation as it will increase the cost and complexity as deploying computational hardware at thousands of individual tower site. Another challenge is to deal with possible connectivity and data protection as the sensor data is moved to access layer for processing and decision-making. So , the local edge computing fit better in IoT/IoE scenarios[18][17].

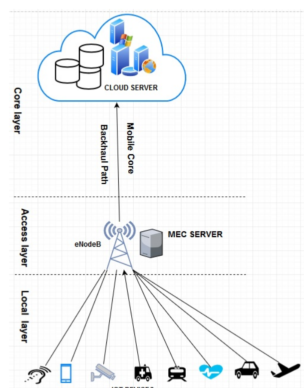

Fig. 2: MEC IoT model.

3) Local Edge Model: Researches proposed different architectures to realize edge computing platforms as scene in figure3. Evaluating these architectures discloses that the edge of the network is not clearly defined and nodes such as gateways expected to connect at the edge can vary. Local network is hop away from the IoT network such as sensors, actuators, mobile devices etc in local edge model. Edge gateway has the ability to compute, store, process the collected data generated by the sensor connected to the edge gateway. Billions of data is generate at the edge layer, therefore, this model is very efficient as compare to IoT-Cloud model to handle the local IoT devices data by providing small cloud services . This model is very much efficient for real time , latency sensitive IoT applications such as video streaming , online gaming , e-health applications etc[15][14].

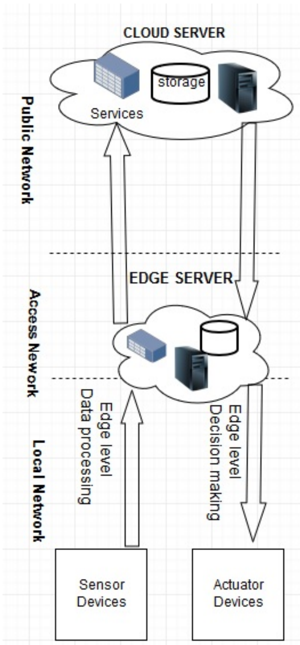

Fig. 3: Local Edge-IoT Model.

4) Tools for Cloud , Fog and Edge Computing: Simulation framework tools are relatively less costly , less complex etc. as compare to construct a test bed in order to evaluate fog and edge computing. There are many simulators used to evaluate cloud computing but fewer are available to evaluate fog and edge computing scenarios[19].

Etemad et al. introduced Discrete Event System Specification(DEVS) simulator in the Fog environment. Nonetheless , DEVS is not useful for simulation IoT devices not even comprises data placement management in the simulated infrastructure[19].

Ruben et al. proposed EmuFog which is a Fog environment emulator. It emulates the switches and routers in the Fog infrastructure. However , this emulator is not available for IoT objects and data placement management emulation. Moreover,it doesnot support mobility also it doesnot support hierarchical structure[20].

Baccelli et al. introduced a test-bed FIT-loT-LAB where thousands of wireless nodes physically located at different sites across France. User can design , services , applications and benchmark IoT protocols using this test-bed experimental platform. Although , in many cases it is too costly and also it does not provide repeatable environment . Furthermore , it does not support Fog environment and data placement[21] .

Gupta et al. introduced iFogSim a simulation toolkit to model the fog environment . Also , compared the resources management techniques in different scenarios in fog computing systems. iFogSim model the structure in hierarchical and there are several structure in iFogSim such as IoT sensors , data stream , actuators ,fog devices etc[16]. In iFogSim allows to simulate real time IoT based applications processing in Fog/Edge environment and measure resource management such as latency , cost , networking congestion , energy consumption etc. in the network[22] .

III. Our proposal

[Describe your simulation model with good figures, what are the centric performance metrics]

In this section , we are comparing with current traditional cloud server model and edge server model with our proposed model. Our proposed model is useful for local level processing and decision making IoT applications such as video streaming , smart traffic light control , smart vehicle transport illustrated in figure 4. For instance , smart traffic light control needs to be processed at the local level and take instant decisions. At peak hours , traffic lights sense the data of how many number of vehicles and pedestrian are cross in that signal, If the number of vehicles are higher than number of pedestrian , the traffic light give more time to the vehicles to cross in order to reduce the traffic congestion vise versa. Also, video streaming is another application which need to be processed locally.

diagram.png

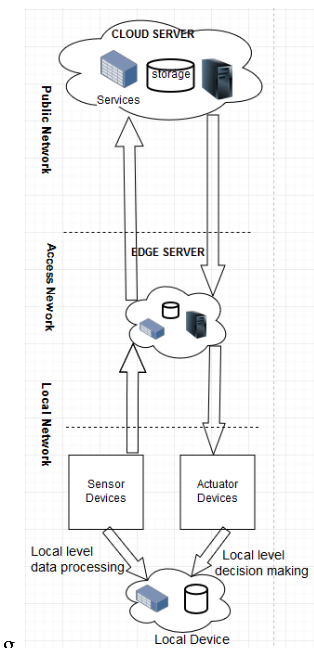

Fig. 4: Local Edge-IoT Model.

Our model helps to reduce the load at access network and traditional cloud server by processing the IoT data and make decisions at the local level as huge amount IoT devices available at local network . The data migration to cloud and access layer for processing are more vulnerable as compared to our local model for IoT platform because the data in servers in the local network is more protective as it exposes to very less public as compared to traditional cloud and access network. Our model provides efficient load balancing than traditional cloud and edge cloud . The number of IoT devices in 2020 reaches to billions which is not an easy to handle by the traditional cloud as smart car will generate 100 Giga byte data per second , each smart mobile device will generate 1 Giga byte data per day, so it leads to network congestion in the traditional cloud server and edge servers in the access layer. Our model helps to reduce the network load at traditional cloud server and edge servers.

The end to end devices sense and generate data all the time which requires a reliable connectivity .Traditional cloud server and edge server are not effective for this purpose. Our model is more reliable in terms of providing end to end connectivity with the local server.

IV. EVALUATION

[Describe the evaluation conditions and results, analyze the results]

In this section, we have evaluated the performance of 3-tier simulation IoT model e.g traditional Cloud-IoT computing , Fog/Edge-IoT Computing and local-IoT computing model .To realize the full potential of local edge computing and IoT paradigms for real time analytics, several challenges need to be addressed in terms of Energy Consumption, number of services executed ,operational cost, Latency , Execution Time.

The results are analysed in iFogSim simulator. iFogSim work in hierarchical manner. Fog devices can be used as cloud server, fog/edge server and local edge server. Each fog device has some parameters such as.

String nodeName Million instructions per second (Mips) RAM Up Bandwidth Down Bandwidth rate per Mips Busy power Idle power Level

A. ENERGY CONSUMPTION

Figure 5. shows the energy consumed at different layers in the simulation. The deployment of application on core layer reaches maximum energy consumption as compared with the access and local layer . As the application deployed on access and local layer fog devices , the energy consumption of the fog devices at the core layer reduces due to data processing and making decision moved towards access and then to local level . On the other hand , the energy consumption of the fog devices increase at the access layer as the processing and decision making is close to the IoT devices instead offloading the data to traditional cloud . Deployment of application on fog devices in local layer compared with the deployment on core and access layer. The energy consumption at local layer increase as the application placement strategy is at one hop from the end to end devices as compared to the other scenario. This is due to the fog devices process the data and make decision at the local layer instead of offload to the access and core layer.

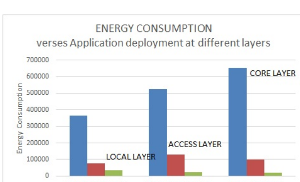

Fig. 5: Energy consumption at different level of MIPS , when application is deployed at core layer.

B. SERVICES EXECUTED

Figure 6. illustrated the number of services executed at different layer . The application placement is deployed at the core layer and compared with the application deployment strategy at access and local layer. It can be seen that the application at the core layer execute more number of services as compared to the fog devices in access and local layer as the fog device in the core layer has more resources and storage capacity than access and local layer fog devices. Also , IoT data offload to the core layer for processing. In the second scenario , the application placement is at the access layer (fog device) which has almost similar function but some how minimum power as compared to the devices in the core layer data centre. It can be clearly seen that the number of services execution are processed increases . On the other hand , the number of services execution in the core layer data centre decrease as some of the services are executed in the access layer which reduces the burden at the core layer due to the application is placed at the access layer . Finally , in the third scenario , when the application is placed more close to the end devices which are connected directly to the fog devices as gateways or Access points. It can be seen that the burden on the core layer in processing the number of services reduced more and it increases at the local layer as compared to the other application placement scenario.

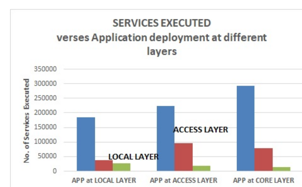

Fig. 6: No. of Executed Services, when application is deployed at different

C. OPERATIONAL COST

Figure7. describe the operational cost compared with the application deployment strategy at each layer . It can be seen clearly for latency sensitive IoT applications such as video streaming , web browsing etc , the operational cost is much more lower than the same application deployed at core layer and access layer as the number of MIPS increases in each scenario. The Operational cost depends on the number of processing nodes , bandwidth , MIPS , RAM etc between the end devices and the core , access and local layer .

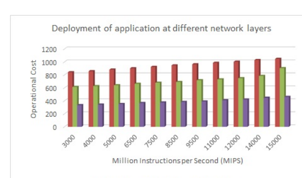

Fig. 7: Operational Cost verses application deployed at different layer.

D. LATENCY

Video latency refers to the degree of delay between the time a transfer of a video stream is re-quested and the actual time that transfer begins. Networks that exhibit relatively small delays are known as low-latency networks, while their counterparts are known as high-latency net-works. Fundamentally, video latency refers to the amount of time it takes for a single frame of video to transfer from the camera to the display. Low latency is typically defined as less than 100 milliseconds (ms) and is preferable when operating remote devices, video conferencing, and streaming live events. High latency, on the other hand, is considered acceptable for applications such as recording and streaming previously recorded events. Network routers are the most notorious devices responsible for latency on the end-to-end path. Satellite communications, on the other hand, often add large amounts of latency as a result of the time it takes a packet to travel across the link. Latency is typically measured cumulatively the more nodes that data must pass through, the more latency there is. For example, the more links and router hops a data transmission must go through, the larger the end-to-end latency can potentially be. Generally, users want latency to be as low as possible. For instance, an operator manipulating a remote device wants to avoid any delay in seeing the delay in frames, audio. etc. Similarly, broadcasters doing live remote interviews do not want delays between speaking and hearing a response.

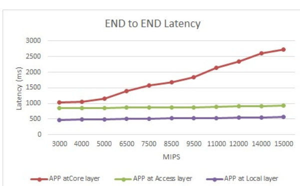

Fig. 8: End-to-End Latency comparison in 3 Tier Scenario vs MIPS

E. TUPLE AND CPU EXECUTION TIME

Figure illustrates the TUPLE and CPU execution time at three different layers such as core , access and loyal layer according to the application placement . As the number of MIPS increases in each scenario the CPU execution time increases accordingly . Here also , local layer shows a promising output in terms of CPU execution time compared with the application deployed at access layer and core layer. CPU execution time depends on the memory consumption and MIPS at the core , access and local layer.

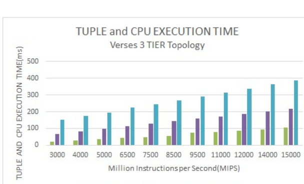

Fig. 9: Execution time , when application is deployed at different layer.

V. Discussion and Future work

[Wrap up the significance of the results, how does your work help scientific community, industry and society]

Acknowledgment

This work was supported by Academy of Finland under projects Industrial Edge, MEC-AI and 6Genesis Flagship (grant 318927) projects.

References

References : [1] https://www.cs.mun.ca/courses/cs6910/IoT-Survey-Atzori-2010.pdf.

[2] CHOY, Sharon WONG, Bernard Simon, Gwendal Rosenberg, Catherine. (2012). The Brewing Storm in Cloud Gaming: A Measurement Study on Cloud to End-User Latency. 1-6. 10.1109/NetGames.2012.6404024.

[3] K. Kang, Z-B. Pang, and C. Wang, Security and privacy mechanism for health internet of things,in The Journal of China Universities of Posts and Telecommunications, vol. 20, no. 2, pp. 6468, Dec. 2013, 10.1016/S1005- 8885(13)60219-8.

[4] M. Liyanage, I. Ahmad, A.B. Abro, A. Gurtov, and M. Ylianttila, A Comprehensive Guide to 5G Security, New Jersey, USA: Wiley Publishing, 2018. [5] M. Liyanage, J. Salo, A. Braeken, T. Kumar, S. Seneviratne, and Ylianttila, Mika, 5G Privacy: Scenarios and Solutions, presented at the 1st 5G World Forum (5GWF), Santa Clara, CA, USA, Jul. 911, 2018. [6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, Edge Computing: Vision and Challenges, in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637 646, Oct. 2016, 10.1109/JIOT.2016.2579198. [7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, Fog computing and its role in the internet of things, in workshop on Mobile cloud computing.ACM, 2012. [8] S. Yi, Z. Qin, and Q. Li, Security and privacy issues of fog computing:A survey, in International Conference on Wireless Algorithms, Systemsand Applications (WASA), 2015. [9] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, The case forvm-based cloudlets in mobile computing, Pervasive Computing, 2009. [10] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, A survey of mobile cloudcomputing: architecture, applications, and approaches, WCMC, 2013. [11] I. Lera, C. Guerrero and C. Juiz, ”YAFS: A Simulator for IoT Scenarios in Fog Computing,” in IEEE Access, vol. 7, pp. 91745-91758, 2019.doi: 10.1109/ACCESS.2019.2927895. [12]M. Aazam, S. Zeadally and K. A. Harras, ”Fog Computing Architecture, Evaluation, and Future Research Directions,” in IEEE Communications Magazine, vol. 56, no. 5, pp. 46-52, May 2018. [13] Puthal, D., Obaidat, M.S., Nanda, P., Prasad, M., Mohanty, S.P., Zomaya, A.Y.: Secure and sustainable load balancing of edge data centers in fog computing. IEEE Commun. Mag. 56, 6065 (2018) Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing [14] A. Khakimov, A. Muthanna and M. S. A. Muthanna, ”Study of fog computing structure,” 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow, 2018, pp. 51-54.doi: 10.1109/EIConRus.2018.8317028. [15]Ashkan Yousefpour, CalebFung, Tam-Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, Jason P. Jue:All One Needs to Know about

Fog Computing and Related Edge Computing Paradigms: A Complete Survey. CoRR abs/1808.05283 (2018).

[16] C. Martn Fernndez, M. Daz Rodrguez and B. Rubio Muoz, ”An Edge Computing Architecture in the Internet of Things,” 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC), Singapore, 2018, pp. 99-102. doi: 10.1109/ISORC.2018.00021

[17] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila and T. Taleb, ”Survey on Multi-Access Edge Computing for Internet of Things Realization,” in IEEE Communications Surveys Tutorials, vol. 20, no. 4, pp. 2961-2991, Fourthquarter 2018. doi: 10.1109/COMST.2018.2849509

[18] Decentralized IoT Edge Nanoservice Architecturefor Future Gadget-Free Computing

[19] M. Etemad, M. Aazam and M. St-Hilaire, ”Using DEVS for modeling and simulating a Fog Computing environment,” 2017 International Conference on Computing, Networking and Communications (ICNC), Santa Clara, CA, 2017, pp. 849-854. doi: 10.1109/ICCNC.2017.7876242

[20]R. Mayer, L. Graser, H. Gupta, E. Saurez and U. Ramachandran, ”EmuFog: Extensible and scalable emulation of large-scale fog computing infrastructures,” 2017 IEEE Fog World Con-gress (FWC), Santa Clara, CA, 2017, pp. 1-6. doi: 10.1109/FWC.2017.8368525 [21] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mit-ton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, T. Watteyne, ”Fit iot-lab: A large scale open experi-mental iot testbed”, 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), pp. 459-464, 2015.

[22] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, iFogSim: A toolkit for model-ing and simulation of resource management techniques in the internet of things, edge and fog computing environments, Softw.: Pract. Experience, vol. 47, no. 9, pp. 12751296, 2017.