International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

Critical Assessment of Phases of Partitioning and Offloading Tasks in Edge Computing

Kulvir Singh, Yogesh Kumar Sharma

Abstract— Edge computing defeats high correspondence defer weakness of customary distributed computing and furnishes figuring administrations with high unwavering quality and high transfer speed for cell phones. At present, edge processing has turned into the front line and hotspot of versatile edge distributed computing (MEC) look into. In any case, with the expanding prerequisites and administrations of versatile clients, offloading procedure of straightforward edge registering is never again relevant to MEC design. This paper advances another astute calculation offloading based MEC design in blend with man-made brainpower (AI) innovation. As indicated by the information size in calculation undertaking of portable clients and the exhibition highlights of edge figuring hubs, a calculation offloading and task relocation calculation dependent on errand expectation is proposed. The calculation task expectation dependent on LSTM calculation, calculation offloading procedure for cell phone dependent on errand forecast, and undertaking relocation for edge cloud planning plan are utilized to help with upgrading the edge processing offloading model.

Keywords— Artificial Intelligence, Computation Offloading, Edge Computing, Task Migration.

I. INTRODUCTION

With late years' expansion of different remote registering applications, for example, wearable gadgets, computer generated reality and cell phones, it prompts a blast of information. Thusly, distributed computing [1] was proposed to give high caliber of administration (QoS) for the applications [2]. In the interim, a few applications additionally require low idleness and the focal distributed computing servers need store and transmit colossal information. These raised downsides of customary distributed computing can be overwhelmed by the Mobile Edge Computing (MEC) [3,4]. The essential thought of MEC is playing out the related undertakings of the application to the close by edges of radio access organize.

The self-sufficient driving which is as of now most concerned subject more often than not including differing sorts of entrusting managing (for example confinement, street representation and course arranging). By utilizing the MEC innovation, cell administrators (the self-ruling vehicles) can satisfy various undertakings effectively.

Still we are going up against numerous difficulties with respect to MEC innovation [5]: (1) calculation offloading; (2) portion of processing asset and (3) versatility the executives. In this paper, we center around the last one. Portability the board is the issue about how to ensure the administration progression for the applications, if the client gear (UEs, we additionally use "client" to point UE) meanders starting with one system locale then onto the next. A few techniques are furnished to handle with UEs' versatility issue. The most immediate strategy is adjusting the developed hub B (eNB) or little cell eNB (SCeNB) transmission control for the offloaded applications. Nonetheless, it must be practical for the UEs with a low portability. With respect to the self-governing driving applications, the virtual machine (VM) movement (or equally, task relocation) model ought to be acquainted with assurance the administrations. Despite the fact that errand movement may invest more energy and backhaul assets for the transmission between MEC hubs, it can present to us the advantages when the self-ruling vehicles experience lower inertness in the vehicles' neighbor and backhaul need not to be doled out for the vehicle. Along these lines, it demonstrates to be an extreme issue to how to boost the relocation increase under the limitation of movement cost (see Fig. 1).

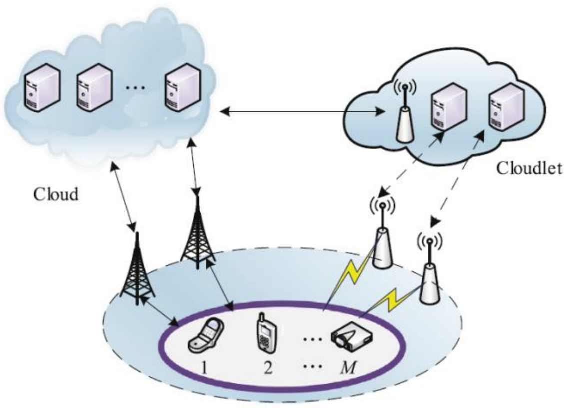

Fig. 1. A system framework for IoT-enabled cloud-edge computing

1.1. Background

Web of Things (IoT) has developed as a mainstream worldview giving internetworking of numerous articles and shrewd things, for example, cell phones and wearable gadgets [1,2].

Manuscript published on November 30, 2019.
* Correspondence Author
Kulvir Singh*, Research Scholar Department of Computer science and Engineering; Shri JJT University Jhunjhunu Rajasthan, India.
Dr. Yogesh Kumar Sharma, Associate Professor (HOD/Research coordinator), Department of Computer science and Engineering; Shri JJT University Jhunjhunu Rajasthan, India.
© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

Presently, the consistently expanding cell phones, inserted with radio recurrence recognizable proof (RFID) and sensor innovation, are associated with IoT through Mobile gadgets in IoT sense the surroundings of portable clients and produce ongoing huge information, with helpful data for supporting the versatile applications [6,7]. The huge information is put away and prepared to ensure the
proficiency and viability of the portable applications. Notwithstanding, the limited calculation limit and reserve size of the cell phones block the wide utilization of the portable applications and cause enormous measure of time for putting away and handling the huge information on the cell phones [8,9]. Additionally, the vitality utilization of the cell phones increments, contracting the life of batteries and increasing emanations of ozone harming substances.

In any case, because of the cloud sent indirectly from the cell phones, offloading the versatile applications to the remote cloud involves significant data transfer capacity of the center system, causing system blockage to a high degree. Thus, EC decreases offloading inertness and makes organize progressively effective with the goal that it gives a timesaving registering worldview [15]. EC empowers a cross breed calculation offloading plan, that is, cell phones can offload the portable applications to the cloudlet or to the cloud. It has the accompanying qualities:

- Located on the edge. The most unmistakable element of edge figuring is that it is situated on the edge instead of center systems. As appeared in the Fig. 2, distributed computing situates on center systems, so information must go through various bounces in systems before arriving at mists. In any case, edge processing situates close to end clients. It is on the fringe of systems.
- Real-time processing. Since it is situated on the edge, it can send information to figuring hubs inside a couple of jumps in systems. Accordingly, it has lower organize idleness and better constant cooperation.
- Extensive geological appropriation. Not at all like distributed computing, edge registering is made out of broadly appropriated sensors and ITDs. For instance, in a brilliant matrix, sensors that make up edge hubs may cover everywhere throughout the city or even the entire nation.
- Support versatility. The broadly disseminated edge hubs enable it to give administrations to moving gadgets. For instance, sensors set along a thruway can give persistent administrations to passed vehicles.
- Heterogeneous gadgets. Gadgets that make up edge figuring may originate from an assortment of makers. Thus, they are frequently extraordinary in item details.
- Unstable systems. Edge figuring is made out of gadgets that may be set up in intense conditions.

Interestingly, remote sensor arrange (WSN) is an appropriated sensor organize, it is made out of various low-control sensors and utilized in military, savvy transportation, natural checking, and so forth [21] However, it can't discover the control result from the hubs. As far as capacity, edge processing is a sort of unmistakable disseminated framework. Nonetheless, the exhibition of this framework is influenced by successive hub notices genuinely, so it needs to repartition these hubs at whatever point hub status is refreshed.

1.2. Inspiration

To improve the presentation of the cell phones in IoT, cloudlets push cloud administrations to the system edge. Versatile applications are frequently formalized as work processes which contain some figuring undertakings with information/control conditions. In cloud-edge figuring, cell phones in IoT are accessible to offload the registering errands to the cloudlet or to the cloud for decreasing the handling inertness and delaying the battery life of the cell phones.

Be that as it may, self-assertively offloading the registering errands barely upgrades the execution time and the vitality utilization of the cell phones, because of the moderate assets of the cloudlet and remote separation of the cloud. Consequently, it stays a test to streamline the execution time and the vitality utilization of the cell phones in the cloud-edge figuring condition. To address the test, a mixture calculation offloading strategy for IoT-empowered cloud-edge figuring is proposed.

Because of the pervasive detecting, processing and joining, Internet of portable things is utilized in a developing number of situations, e.g., medicinal services framework and providing food business [3,4]. To render versatile clients improved encounters and increment the administration nature of cell phones, the Internet of portable things encourages rich portable applications, including estimating commotion, recording area and catching pictures [5].

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this work, we consider a framework model as appeared in Fig. 1. A gathering of N versatile clients offload their calculation undertakings to the edge server through remote connections. Every client has M free errands to be finished.

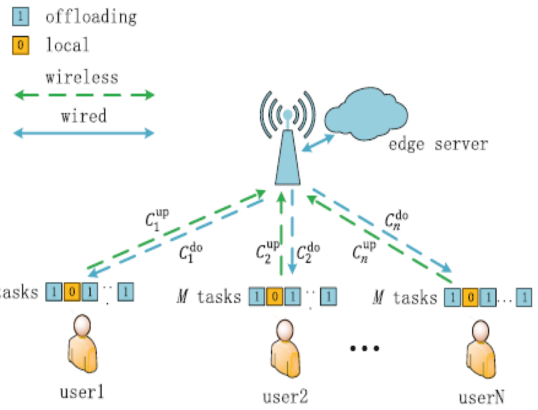

Fig. 1. System model

To show the assignment, we use Lin nm to mean the info information size of client n's mth task, i.e., the remaining burden of undertaking m which should be transmitted from client n to the edge server. Furthermore, we use Lout nm to signify the yield information size of client n'smth task, i.e., the remaining burden of taskm which should be transmitted once more from the edge server to client n. In the interim, every client n can decide if to offload its mth errand to the edge server or not.

At that point we utilize the twofold factor xnm ¼ 1 to signify that client n chooses to offload its errand m to the edge server, and xnm ¼ 0 implies that client n chooses to execute its undertaking m locally. We'll give a point by point displaying of the calculation offloading in the following subsections. Table 1 records the significant documentations utilized in this paper.

2.2. Cost of calculation offloading

At the point when clients offload their undertakings to the edge server, the vitality utilization for client n's assignment m incorporates two sections, i.e., the part for information transmission and the part for information gathering. Specifically, we signify the vitality utilization for information transmission as Et nm and the vitality utilization for information gathering as Er nm. Notwithstanding the vitality utilization of versatile clients, we likewise utilize a comparative methodology as [5] to demonstrate the framework utility expense of the edge server.

Notice that the expense Eclonm incorporates the vitality utilizations for sending and getting the undertaking and the server's utility expense for executing this errand.

Next, we model the postponement in calculation offloading. In particular, we use Cup n to mean the assigned transmission capacity to client n for transmitting its offloaded errand to the edge server and use Cdon to indicate the allotted transfer speed to client n for getting the yield information of its undertaking from the edge server.

2.4. Issue detailing

One answer for take care of the primary issue is to utilize substitution calculations to supplant new information with past information saved money on current hub. The center issue of a substitution calculation is the way to choose which parts of information to be expelled from current edge hub and transferred to upper stockpiling levels when capacity asset isn't sufficient. A substitution calculation should keep much of the time utilized information in inclination to never utilized or inconsistently utilized information in current hub. In the event that regularly utilized information is transferred to upper levels, the framework will recover information from other stockpiling levels as often as possible, in which a program will invest more energy perusing information contrasted and the case that much of the time utilized information is kept in current hub. In this way, an appropriate substitution calculation is basic to upgrade framework execution. Actually, the accompanying issues must be comprehended for an effective substitution calculation.

Substitution factors

There are numerous variables that have an effect on framework execution. As referenced in the Section 2, FH is a significant factor. Records and information with high FH ought to be kept in current stockpiling hub while that with low FH ought to be transferred first. Aside from FH, numerous different variables can likewise influence framework execution. To address the second issue that systems are not steady, the significance of information ought to likewise be considered. On the off chance that information with high significance is supplanted and transferred to upper stockpiling levels first, the loss of significant information will be decreased when edge hubs are lost.

Substitution estimation

Substitution estimation gauges whether a document ought to be expelled. In the wake of deciding the substitution factors, a recipe for figuring it ought to be given.

Information structures

A framework needs to record additional data like access recurrence, document size and other related qualities to actualize a substitution calculation. This data not just involved a piece of capacity assets of the edge hubs, yet in addition requires a piece of registering assets to oversee and process it. In this manner, it is essential to plan proficient information structures for substitution calculations. For a certain something, effective information structures spare stockpiling assets of the edge hubs. For another, they can improve calculation productivity and spare figuring assets.

Capacity association

A legitimate stockpiling association is useful to improve the perused and compose productivity of a capacity framework. It is increasingly significant for edge registering due to the restricted extra room of the edge hubs. Accordingly, the association of free space and the executions of information structures are basic issues to plan a productive substitution calculation.

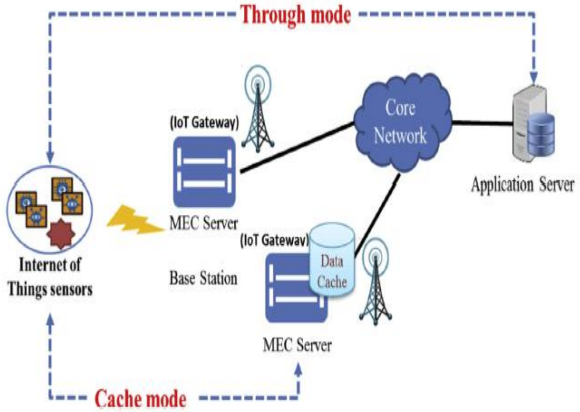

Fig. 3. Traffic offloading modes

III. WORK DONE

There is an expanding enthusiasm being appeared by the worldwide research network on VM designation and relocation procedures to diminish the power utilization in distributed computing and edge figuring conditions. The momentum examines and audits are drawing the consideration of scientists and experts towards the power utilization issue relating to the overabundance underutilized dynamic servers in the cloud server farms.

Countless existing works have considered the advancement issues on portable edge processing. A large portion of these works center around the basic leadership issue for errand offloading at the client end. These works will initially evaluate the normal postponement or vitality utilization for objective assignments and after that offload them to the portable edge if the deferral or vitality utilization is decreased. As indicated by the improvement objective, we partition these works into two classifications: chips away at streamlining vitality utilization and deals with decreasing the undertaking delay.

Our work falls into the subsequent classification and contrasts from the current works in that our point is to lessen the postponement at the system end rather than the client end.

Takes a shot at advancing vitality utilization.

K. Zhang et al. [17] displayed a multi-gadget calculation offloading structure for MEC and detailed an enhancement issue that limited the gadget vitality utilization. A three-arrange offloading plan was proposed to get the imperfect arrangement, which (1) characterized the cell phone, (2) decided the need and (3) assigned the radio asset. X. Chen et al. [9] further considered the impedance and impacts when there were such a large number of clients attempting to offload undertakings to similar sBS, which can essentially build the vitality utilization of cell phones. The offloading was defined as a multi-client game, which was demonstrated continually conceding a Nash balance.

W. Labidi et al. [18] considered the time shifting channel state for remote offloading and proposed a booking plan for undertaking offloading, which attempted to utilize remote channels and client cushions to diminish the vitality utilization. Deals with diminishing the assignment delay. J. Liu et al. [11] attempted to limit the execution delay for single clients with one-dimensional inquiry calculation. The calculation yielded a strategy for offloading choice as indicated by the application cradle lining state. In addition, the attributes of remote channels were additionally considered. Y. Mao et al. [19] mutually enhanced the errand offloading booking and transmitted power allotment issue to lessen the offloading delay. Plachy et al. [20] considered the spatial decent variety of the sBSs in the offloading procedure. The sBS that was in charge of undertaking execution was picked by the clients. At that point the outcomes would be come back to the clients through the sBS with the most elevated RSSI of the remote connections. Be that as it may, the work was intended for single undertaking offloading.

There are likewise a few works that together streamline both vitality and undertaking delay. Diminishing errand delay frequently includes extra vitality utilization in MEC, particularly for the assignments that execute quicker on cell phones than on versatile edges. A few works set a vitality edge and after that limit the execution delay without surpassing the vitality edge. For instance, Y. Mao et al. [14] proposed a dynamic offloading plan for single client to limit the execution delay for vitality reaping gadgets, where the vitality gathering strategy added intricacy to the offloading calculations.

A lightweight guess approach was proposed to accomplish a decent tradeoff among multifaceted nature and the postpone minimization. J. Yang et al. [21] utilized a multi-arrange consecutive game model to meet the vitality and defer prerequisites simultaneously. Not quite the same as the previously mentioned works, our work stresses on limiting the postponement inside the MENs. The deferral in MENs not just adds to the general undertaking execution delay yet in addition impacts the basic leadership process at the client end. We propose a novel in-organize errand planning way to deal with diminish the assignment execution delay, where undertaking data, asset data and client versatility are together considered. Not quite the same as the [13] which utilized the contact rate to catch client portability, our work is based over the client direction forecast [15,16], which can precisely catch client versatility in most MEC situations, for example, air terminals, shopping centers, and so forth. In addition, to manage the effect on offloading choice, we propose a novel defer estimation conspire for cell phones.

Kansal et al. [12] features that asset designation is a major piece of asset planning. All the more explicitly, the creators manage the enormous scale cloud server farms. Be that as it may, the arrangement isn't pertinent for all size of server farms. A tattle convention is planned by Yanggratoke et al. [31], in which the changing of burden example is taken as a criteria for the solidification of the VMs. Be that as it may, it is a heuristic arrangement and the creator implies that it works productively in specific conditions and may not be material to a wide range of situations. Jha et al. [11] presented a heap adjusting calculation for Infrastructure as a Service Cloud to alleviate the power utilization in the cloud server farms. Be that as it may, the computational time isn't examined subsequent to executing the heap adjusting methodology. Hereditary calculations are development based calculations and are said to give better outcomes after a specific number of ages. Ranjbari et al. [21] proposed a learning automata based calculation to upgrade the asset usage and alleviate the vitality utilization. In any case, the proposed technique is an expectation approach contingent upon the adjustments in the client's asset requests.

A period arrangement estimating approach is displayed by Zhou et al. [33] for the non-deterministic outstanding burden condition when there is no from the earlier data about the approaching occupations. By the by, a limit explicit to the reproduction condition is utilized by the proposed framework to change in accordance with the dynamic outstanding task at hand. Wang et al. [25] proposed Ada-Things to screen and empower live movement strategy for the IoT application in the edge cloud engineering. The outstanding task at hand data is the fundamental criteria adjusted for relocation to accomplish a reasonable parity in the exhibition. Hereditary programming based and swarm insight based streamlining models present disadvantages like high blunder rate and low practicality notwithstanding the overall versatility challenge [3].

A vitality proficient arrangement with expectation of up and coming employments dependent on Gaussian procedure relapse strategy is proposed by Bui et al. [5]. Be that as it may, a thorough methodology is adjusted with a supposition that bound number of physical servers can figure out how to serve the approaching VM demands.

The paper [6] figured the issue about VM relocation as a nonstop time Markov Decision Processing (MDP) and plan an arrangement to decide if starting the VM movement. The [7] additionally use MDP to manage the VM movement yet it generally starts the VM relocation as long as the UE is limited by the edges.

The primary disadvantage for [6] and [7] is that their proposed strategies all depend on one-measurement models. As a matter of fact, the general setting for the VM movement includes 2D portability. The paper [8] improve their VM relocation process by a portability expectation: gauge a propelled throughput when wandering all through the system.

IV. FRAMEWORK

As presented before, the target of this paper is to propose an effective offloading strategy that chooses which errands ought to be offloaded and to which offloading server (cloudlets or cloud), while limiting the absolute vitality devoured by the mobiles. Given our framework depiction and as per the QoS and offloading servers' assets abilities requirements, our concern can be defined as pursues:

Minimize $\sum_{m}^{M}\sum_{n}^{Nm} Zm,n$

As showed over, our goal is to limit the aggregate sum of vitality devoured by the mobiles. Here $X^{k}{}_{m,}$ is the offloading choice of the assignment of the client (m, n) to the offloading server k, which implies that $xkm,$ = 1 if the client (m, n) offloads its undertaking to the offloading server k, and 0 generally. Zm,n is the measure of vitality devoured by the errand of the client n on the AP m

V. DISCUSSION & CONCLUSION

In suggestion to the inadequacies of customary nearby figuring, edge registering and distributed computing modes, another shrewd calculation offloading based MEC design is advanced with mix of three modes in this paper. We additionally examine the examination purposes and points of interest about the new age of MEC. In view of the proposed engineering, the calculation offloading and task relocation calculation dependent on undertaking expectation is planned. The ideal calculation offloading methodology is portrayed in detail from viewpoint of LSTM-based calculation task forecast calculation, task expectation based registering offloading procedure for cell phones, and calculation task relocation for edge cloud planning plan. In view of our proposed calculation and design, execution tests are led. Through examination with neighborhood figuring and single edge registering offloading procedure, our calculation can adequately decrease the all out errand defer when the calculation information sum and subtask sum increment, in this way to guarantee effective handling of time-postpone delicate assignments.

These days, Internet of versatile things has developed as a prevalent innovation for realizing rich portable applications. With the advancement of the innovation, the intricacy and sizes of the huge information for procedure increment, which has clashes with the asset constraint of cell phones. Moreover, broad investigations and assessments are directed to avow the proposed strategy COM performs well in taking care of the enhancement issue.

For future work, we will modify and expand the proposed technique in a genuine situation of IoT. Moreover, we will take shape diverse time prerequisites of the work processes for execution, attempting to discover an offloading system to accomplish the most extreme vitality utilization reserve funds of the cell phones.

REFERENCES

1. M. Armbrust et al., “A view of cloud computing,” Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.
2. S. Ghemawat, H. Gobioff, and S.-T. Leung, “The Google file system,” ACM SIGOPS Oper. Syst. Rev., vol. 37, no. 5, pp. 29–43, 2003.
3. J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,” Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.
4. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in Proc. IEEE 26th Symp. Mass Storage Syst. Technol. (MSST), Incline Village, NV, USA, 2010, pp. 1–10.
5. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster computing with working sets,” in Proc. 2nd USENIX Conf. Hot Topics Cloud Comput., vol. 10. Boston, MA, USA, 2010, p. 10.
6. K. Ashton, “That Internet of Things thing,” RFiD J., vol. 22, no. 7, pp. 97–114, 2009.
7. H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, “Vision and challenges for realising the Internet of things,” vol. 20, no. 10, 2010.
8. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Future Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.
9. “Cisco global cloud index: Forecast and methodology, 2014–2019 white paper,” 2014.
10. D. Evans, “The Internet of Things: How the next evolution of the Internet is changing everything,” CISCO White Paper, vol. 1, pp. 1–11, 2011.
11. D. Evans, “The Internet of Things: How the next evolution of the Internet is changing everything,” CISCO White Paper, vol. 1, pp. 1–11, 011.
12. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 68–73, 2008.
13. E. Cuervo et al., “MAUI: Making smartphones last longer with code offload,” in Proc. 8th Int. Conf. Mobile Syst. Appl. Services, San Francisco, CA, USA, 2010, pp. 49–62.
14. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for VM-based cloudlets in mobile computing,” IEEE Pervasive Comput., vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.
15. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of things,” in Proc. 1st Edition MCC Workshop Mobile Cloud Comput., Helsinki, Finland, 2012, pp. 13–16.
16. Boeing 787s to Create Half a Terabyte of Data Per Flight, Says Virgin Atlantic. Accessed on Dec. 7, 2016. [Online]. Available: https://datafloq.com/read/self-driving-carscreate-2-petabytes-data-annually/172
17. Self-Driving Cars Will Create 2 Petabytes of Data, What are the Big Data Opportunities for the Car Industry? Accessed on Dec. 7, 2016. [Online]. Available: http://www.computerworlduk.com/news/data/boeing-787screate-half-terabyte-of-data-per-flight-says-virgin-atlantic-3433595/
18. Data Never Sleeps 2.0. Accessed on Dec. 7, 2016. [Online]. Available: https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/[19] (2016). OpenFog Architecture Overview. OpenFog Consortium Architecture Working Group. Accessed on Dec. 7, 2016.
19. [Online]. Available: http://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf
20. S. Yi, Z. Hao, Z. Qin, and Q. Li, “Fog computing: Platform and applications,” in Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb), Washington, DC, USA, 2015, pp. 73–78.
21. K. Ha et al., “Towards wearable cognitive assistance,” in Proc. 12th Annu. Int. Conf. Mobile Syst. Appl. Services, Bretton Woods, NH, USA, 2014, pp. 68–81.
22. B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “CloneCloud: Elastic execution between mobile device and cloud,” in Proc. 6th Conf. Comput. Syst., Salzburg, Austria, 2011, pp. 301–314.
23. A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, “Saving portable computer battery power through remote process execution,” ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 2, no. 1, pp. 19–26, 1998.
24. G. C. Hunt and M. L. Scott, “The coign automatic distributed partitioning system,” in Proc. OSDI, vol. 99. New Orleans, LA, USA, 1999, pp. 187–200.
25. K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” Computer, vol. 43, no. 4, pp. 51–56, Apr. 2010.
26. S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in Proc. IEEE INFOCOM, Orlando, FL, USA, 2012, pp. 945–953.
27. L. Zhang et al., “Named data networking (NDN) project,” Xerox Palo Alto Res. Center, Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010.
28. D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, “MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet,” ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 16, no. 3, pp. 2–13, 2012.

29. Dr. Yogesh Kumar Sharma and S. Pradeep (2019), “Deep Learning based Real Time Object Recognition for Security in Air Defense”, “Proceedings of the 13th INDIACom; INDIACom-2019; IEEE Conference ID: 46181 2019 6th International Conference on “Computing for Sustainable Global Development”, 13th - 15th March, 2019 Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), IEEE - New Delhi, (INDIA)”, ISSN No: 0973-7529 , ISBN: 978-93-80544 , Volume : 32, Issue: 8, Pp. 64-67.
30. J. Feld, “PROFINET—Scalable factory communication for all applications,” in Proc. IEEE Int. Workshop Factory Commun. Syst., Vienna, Austria, 2004, pp. 33–38.
31. J. Cao, L. Ren, W. Shi, and Z. Yu, “A framework for component selection in collaborative sensing application development,” in Proc. 10th IEEE Conf. Coll. Comput. Netw. Appl. Worksharing, Miami, FL, USA, 2014, pp. 104–113.
32. F. DaCosta, Rethinking the Internet of Things: A Scalable Approach to Connecting Everything. New York, NY, USA: ApressOpen, 2013.
33. WiFi Network Security Statistics/Graph. Accessed on Dec. 7, 2016. [Online]. Available: http://graphs.net/wifi-stats.html/
34. Dr. Yogesh Kumar Sharma and Dr. Surender (2013), “Future Role of Zigbee Technology in Wireless Communication System", Paper published in Grip - The Standard Research International Referred Online Research Journal, ISSN-2278-8123, Issue No. XVI, Pp. 18-31.
35. Open Mhealth Platform. Accessed on Dec. 7, 2016. [Online]. Available: http://www.openmhealth.org/
36. K. R. Jackson et al., “Performance analysis of high performance computing applications on the Amazon Web services cloud,” in Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom), Indianapolis, IN, USA, 2010, pp. 159–168.
37. A. Li, X. Yang, S. Kandula, and M. Zhang, “CloudCmp: Comparing public cloud providers,” in Proc. 10th ACM SIGCOMM Conf. Internet Meas., 2010, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879143
38. M. Satyanarayanan, “Mobile computing: The next decade,” SIGMOBILE Mobile Comput. Commun. Rev., vol. 15, no. 2, pp. 2–10, 2011. [Online]. Available: http://doi.acm.org/10.1145/2016598.2016600
39. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 68–73, Dec. 2008. [Online]. Available: http://doi.acm.org/10.1145/1496091.1496103
40. M. Armbrust et al., “A view of cloud computing,” Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672
41. A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in Proc. 2nd USENIX Conf. Hot Topics Cloud Comput., Boston, MA, USA, 2010, p. 4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863107
42. N. Ding et al., “Characterizing and modeling the impact of wireless signal strength on smartphone battery drain,” SIGMETRICS Perform. Eval. Rev., vol. 41, no. 1, pp. 29–40, Jun. 2013. [Online]. Available: http://doi.acm.org/10.1145/2494232.2466586.