Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

An Analysis Of Edge-Cloud Computing Networks For Computation Offloading

Naga Lakshmi Somu[1], Dr Prasadu Peddi[2]

[1]Research Scholar, Department of Computer Science, Shri Jagdishprasad Jhabarmal Tibrewala University, Rajasthan.

[2]Assistant Professor, of Computer Science, Shri Jagdishprasad Jhabarmal Tibrewala University, Rajasthan.

ABSTRACT

The proliferation of large amounts of data brought about by the Internet of Things in the commercial and academic spheres has resulted in a significant rise in the number of cloud data centers that provide data analytics services. As a result of the persistent and pervasive requirement to analyze data in close proximity to its original source, edge computing has become more common in recent years. This is one of the reasons why edge computing has become so popular. As a result of edge computing, the processing load on the network's periphery as well as the data centre may be reduced. In addition to this, it is more private and may better accommodate the requirements of the service providers. Edge computing offloads processing by dynamically distributing workloads between a cloud data centre, edge servers, and an edge device. This helps edge computing improve the transmission of network traffic and increases the responsiveness of the system. In this study, we undertake a thorough literature analysis in order to demonstrate how far we have come in the field of computational offloading to edge computing. Specifically, we want to illustrate how far we have progressed in the area of edge computing. Analyses are conducted on a number of elements of offloading computation, including its influence on energy usage, service quality, and customer happiness. Methods for allocating resources are covered, such as gaming, along with methods for balancing system performance and overheads while offloading computations.

Keywords: Task partitioning, Offloading, Edge computing, computation offloading, Optimization, edge-cloud collaboration.

1 INTRODUCTION

A cloud-based computing system is system to provide services, which gives clients access to distributed and scalable computing capabilities, which include storage, networking and computing within cloud-based data centers. Cloud Service Providers (CSPs) offer flexibility and efficiency for their customers through offering services like software as a service (SaaS) or platform as a service (PaaS) and infrastructure as a service (IaaS). For instance, they can expand or reduce the amount of their services depending on their clients requirements, develop customized applications and access cloud services from any place connected to the internet. Cloud-based solutions are perfect for companies with increasing or changing demands for internet bandwidth. Furthermore, by using cloud computing, enterprises can launch their applications faster, without having to worry about the cost of infrastructure maintaining, recovery and automated software updates. To maximize the advantages that cloud computing offers, different deployment models , such as the private cloud as well as public cloud and hybrid cloud are essential to ensure the reliability of systems and their capacity to meet the business requirements. Since its beginning cloud computing has transformed the business model across all verticals as well as the human everyday life in a dramatic way. In addition, the corporate IT investments in cloud-based services will be more rapid than that of traditional (non-cloud) IT offerings, this move to software-only solutions makes cloud computing among the most disruptive elements within IT markets. It is therefore expected that cloud computing will end up increasing in popularity and will be a common feature within every commercial or personal market, which is in line with the increasing use of the Internet in the present.

Computational offloading was thought of as a means to overcome the limitation in resource that mobile phones have. Computational offloading can be described as the method of sending a task as well as the associated data to distant high-end resources such as clouds servers and edge servers for processing according to the requirements [2]. Many factors impact the decision to offload, including local device resources like storage, CPU and battery usage, along with the amount of time needed for processing through the network, as well as the volume of transferred data [3].

The concept of edge computing that was first proposed that brings the storage and computation closer to the users Environments UEs that was that was proposed in 2009 is the cloudlet. The concept behind cloudlet is to put computers with high computing power at strategically placed spots to provide computational resources as well as storage for UEs within the vicinity. The cloudlet concept of cloudlet concept of computing "hotspots" is similar to WiFi hotspots but instead of Interne internet connectivity, cloudlets provide cloud-based services to mobile users. Cloudlets are expected to be accessible to mobile UEs by WiFi connectivity is viewed as a sign of

TABLE I: High level comparison of Edge computing concepts

Technical aspect	Edge Computing	Cloud	Cloud computing
Deployment	Centralized		Distributed
Distance to the UE	High		Low
Latency	High		Low

Jitter	High	Low
Computational power	Ample	Low
Storage capacity	Ample	Low

Varghese and Buyya [1] Examine the evolution and development of cloud computing. We begin with cloudlets, ad-hoc clouds multicloud, heterogeneous clouds and multicloud. Micro-clouds are presented, as well as four different cloud computing models including fog computing and mobile edge computing, as well as the use of volunteer computing in addition to software-defined computing and serverless computing. They also discuss the possible consequences that cloud computing can be bringing to IoT (Internet of Things) and big data-driven autonomous learning systems. They also analyze the challenges involved in designing a cloud computing technology that is suited to the improvement of reliability and security in a long-lasting cloud system, as well as efficient methods for managing resources.

2 COMPUTATION OFFLOADING

Computation offloading allows you to run applications that are extremely demanding remotely to gain advantages from the the power of cloud (or the edge) servers that are able to outlast CPU and battery limitations on the mobile side.

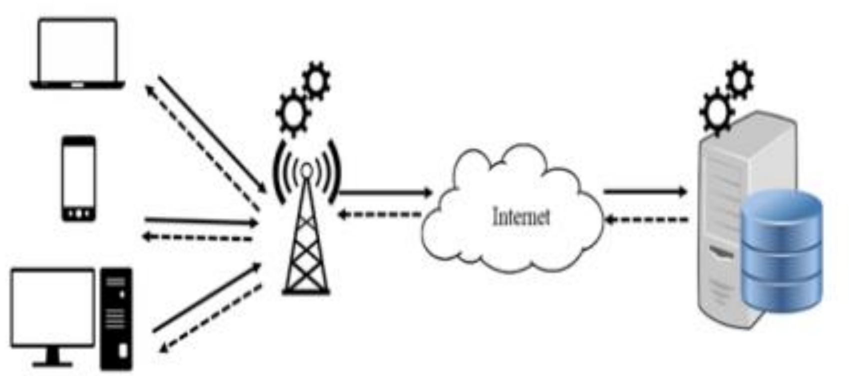

Fig 1: Offloading architecture

Offloading of computation can be either dynamic or static. The static offloading process first evaluates the performance of offloading with offline profiling or models for performance estimation. The load-offloading dynamic process starts by conducting a static examination of the source code and other resources. When the application is running, it is examined dynamically, which allows it to split applications into two main components.

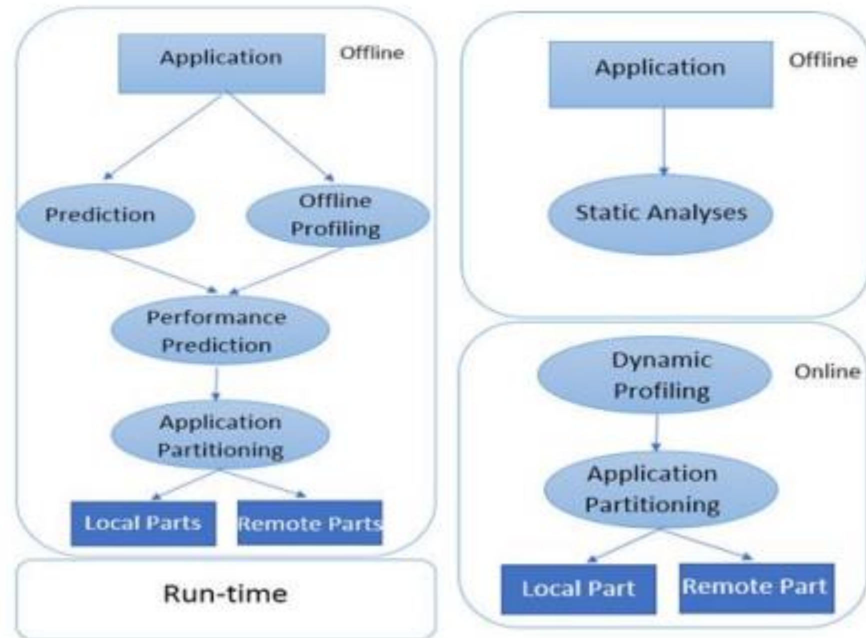

Fig 2: Static and dynamic offloading - (a) Static offloading (b) Dynamic offloading

Computation offloading is considered to be an efficient method to ensure high-quality service to the user by delegating the tasks that require a lot of computation or are latency sensitive to edge devices or adjacent edge servers [12].The main reason behind the offloading of computations is that it helps reduce the time it takes to react to the service, and to improve the quality of service. Additionally, if the edge node isn't equipped with the capacity to complete this task it could be moved into a cloud or edge data centre to boost the performance of the whole system. When deciding to offload computations , a variety of factors must be taken into consideration such as efficiency improvement and energy efficiency reduction. There are many problems to be considered prior to the offloading of computations:

(1) Does the task be delegated? The task scheduler needs to determine whether the task is able to be delegated, i.e., what is the best method to offload it, a partial or full offload?

(2) What is the best time to delegate the task? The task scheduler has to determine the appropriate time for offloading in accordance with various limitations.

(3) Which location should you offload? The answer is what is the ideal location for the offloading of workloads in accordance with the available resources.

(4) Which policy for offloading will be implemented? What is the principal goal of offloading of workloads and single performance metric maximization or joint optimization or tradeoffs between multiple goals? For example, massive edge devices differ in terms of their architectural performance indicators, performance and power supply modes which result in different energy efficiency distributions between devices. In addition, dynamic changes to the bandwidth and latency of networks between clouds data centers and edge equipment can result in changes in

energy consumption for data transmission. So, a diverse policies for offloading computations result in different energy consumption. So, a sound algorithm for offloading computations must determine the most effective balance between total time delay for computation, the data transmission, as well as the related performance indicators.

Within this segment, we'll look at the research that is currently being done on computation offloading. We will also discuss some issues and possible research direction.

3 WHAT TO OFFLOAD THE SELECTION OF THE OFFLOADED WORKLOAD

Edge computing's promise to cut the latency of applications and consumption of bandwidth will not be realized until the cloud data centers are transfered to the edge device or servers. In the case of an cloud-edge co-existing system the primary tasks being executed by cloud data centers need to be split, and some tasks should be relegated to run on edge devices, as and edge servers. In certain instances like IoT the local processor can be utilized dispersed across a large variety of devices and local computing resources aren't sufficient to handle the demands of complicated applications. Therefore, a careful allocation of workloads offloaded to devices at the edge will aid in achieving lower latency as well as better system performance.

Locally caching content is the most popular method of speedier delivery of content across a broad range of applications. Similar to cache data coming from cloud data centers, transferring it to nearby edge devices as well as near edge server may provide less latency for the delivery of content. For instance, modern web servers have lots of dynamically-generated pages , and dynamic websites dominate web traffic, especially ones that have dynamic content such as streaming music or streaming videos.

To improve the quality of the user satisfaction (QoE) by making data available to users, and offloading the processes to proxy servers or by caching portions of dynamic pages , as well as creating pages with compositions upon access to the page are all effective techniques. Yuan et al. [13] suggested not to transfer the central information to the clients and also offloading or caching on the edge server in order to decrease the speed of application. In addition, filtering the majority of server requests into web proxy servers can significantly decrease the server's load. Chen et al. [14] introduced the idea of using network caching to make use of the storage capacity of multiple networks in order to reduce traffic on networks.

3.1 WHEN TO OFFLOAD: THE PRECISE TIMING

As conditions on the network are constantly changing as applications run, the any decision regarding workload offloading has to determine the time when the load should be delegated. Also the task scheduler needs to precisely determine the best time to offload with regard to the system's state and conditions. For instance data caching when networks are congested can enhance the performance of the system as well as transferring large amounts of transfer of data to cloud data center is feasible in the event that the connection to the cloud data centers is adequate to allow data communications. In the last section we have discussed the subject of selecting an offloaded task like data caching, the storage of data and offloading processing and analysis. In this article , we will look at the method of addressing the question of when it is appropriate to offload. The question as to when to take offload can be transformed into a question of when exactly at what times the offloading of workloads will yield the greatest productivity gains while reducing costs, like the consumption of energy and bandwidth. When the offloading of computation is selected the task and the data are divided into smaller chunks. Due to the dynamics of the connections to networks and the accessibility of edge devices the precise timing of workload offloading is crucial for more system performance and reduce use of resources. Furthermore, the order in which execution is performed by the partitioned workload may have an consequences on system performance. Monitoring system performance and the characterization of workloads such as task arrival rates and deadlines could help in making an informed decision about offloading.

3.2 WHERE TO OFFLOAD: THE SCHEDULING OF OFFLOADED WORKLOADS

The offloading of the work can be accomplished by assigning of partitioned tasks to specific edge server and devices. The selection of the appropriate edges servers and edge equipment demands multi-objective optimization, which includes the consumption of energy and the performance of bandwidths, bandwidths for networks and a data privacy security method. In this case an inherent scheduling rule uses energy. Tasks are transferred to cloud servers to save energy. Meanwhile, the tasks that require data are moved to edge servers so that they can provide lower latency and less use on the network. In particular offloaded task scheduling needs to be based on the overall system state including the status of the network, task requirements and device information, for example. If, for instance, the bandwidth of the network is adequate cloud servers could be selected for the execution of workloads and edge servers, as well as local machines are the best locations for the execution of workloads. Additionally, if the work needs low latency, edge servers are the best place to allow task execution.MpOS [8]is among offloading frameworks which complete all tasks that are related to the decision to offload on mobile devices using a the decision tree. The MpOS framework has the potential to reduce power consumption for mobile phones by using the adaptive monitoring method and can lower energy consumption by as much as 55 percent.

3.3 ENERGY AND QOS TRADEOFF BETWEEN COMPUTATION AS WELL AS DATA TRANSMISSION

Today increasing numbers of applications run on smart mobile devices. The quality of the user's experience is the most important method to determine the efficiency of apps and devices. However, mobile phones situated in the middle of networks typically aren't equipped with sufficient resources, such as capacity for storage and computing and battery capacity, making it challenging to satisfy the increasing demands that mobile customers have. To provide better standard services, these resources need to be assigned and scheduled in accordance with the requirements of users and service levels agreements (SLAs). Thus, applications that are sensitive to delays should be prioritised, and those that require computation should have sufficient computing resources. To attain this,"quality of service" (QoE) is an individual's perception of quality of service as well as the performance of the devices, systems or networks as well as software. The transfer of calculations to servers at the edge and returning the outcome of the computation back to mobile devices could drastically reduce the requirement for the resources available to the mobile phone. In order to offload computations, you must to comply with QoS requirements and QoE. It is also necessary to create an appropriate schedule for offloading tasks and determine the time of offloading of each project.

4 COMPUTATIONAL OFFLOADING CHALLENGES AND ISSUES

This section will focus on the main issues that arise in MEC environments through the concept that offloads computation.

Resource allocation: One of the toughest issues to solve. It is determining the amount of resources needed to complete an operation. If the resources aren't sufficient then the offload procedure is the one that will be most preferred. If the available resources are greater than the amount that needed for service then the system won't be used [15].

Scalability: Real-time applications typically run by multiple users at once which are controlled by different algorithms responsible for the multitude of demands of users. In order to allow the app to expand without harm, it is essential that the method of offloading needs to be improved in order to ensure that the process of offloading is improved [16].

Security: In the process of offloading, tasks as well as user's information are transferred through the network, increasing the possibility of theft and misuse of data. To minimize the risk, it is essential to work with a reputable company once the decision to sell off is made [15].

A tradeoff in energy usage Offloading itself is an energy-intensive procedure that requires energy and bandwidth, therefore the decision to decommission should be made in line with the trade-offs [16].

Making decisions: It can be an issue to decide when to let a job go due to a myriad of variables like delays, energy consumption or even payment costs [8].

Accessibility Mobile devices have to be connected to the cloud/edge constantly and this is often difficult due to the insufficient coverage of networks, network congestion, the deficiency of bandwidth, and many other issues with networks. [16].

Mobility: This is a an issue for connecting the device to its edge[19].

Load balancing: Edge nodes and datacentres can get overwhelmed any moment. This can result in delays, and in some circumstances, result not accurate. To prevent this from happening load-balancing is employed to divide tasks between other edge nodes which are underloaded [15]. The load-balancer is a tool that helps to share tasks with other edge no.

Utilization of resources: The number of resources available at the edge is limited when compared with cloud-based infrastructure. Therefore, a management of resource utilization is essential to reap the maximum benefits from limited resources while avoiding excessive system burden [20].

5 MAIN COMPUTATIONAL OFFLOADING FRAMEWORKS

This section focuses on the major computational offloading systems that are being developed by researchers to improve various parameters like time and energy, scalability, and so on. Some are examined in relation to edge while others are explored with respect to computing.

Zhang et al. [2] A theoretical framework was proposed to cut down on the energy use based upon wireless conditions parameters that can be used for applications that make decisions to offload. They employed a threshold policy that was based on the wireless transmission model, as well as the energy coefficients ratio of computations in both cloud and mobile environment.

Orisini et al. [5] suggested the CloudAware framework to offload tasks to devices that can perform edge computing. They suggested that their CloudAware framework could allow for real-time and ad-hoc communication and would integrate computation offloading and contextual adaptation based on the experience of the developers and the component scheme to take the decision to offload. However, they presented their design without implementing or testing.

Habak et al. [7] The Femtocloud system was invented by The Femtocloud technology, that made use of the capabilities of mobile devices to establish a cooperative computation cluster in solid environments, such as coffee shops , classrooms, and other places where people frequently meet. The framework showed the advantages of being more flexible as compared to the cloudlet. Furthermore it was built on unfixed, cooperative services in the edge layer that make up the networks. The analysis conducted by the creators of their method was conducted on tests and simulations , and proved that the system worked with regard to energy consumption and computation. It also reduced the time to complete computation.

Huang et al. [9] looked into wireless-powered MEC networks. They have developed a deep learning online offloading algorithm called DROO which divides an optimization problem into two distinct problems including how to allocate resources as well as the decision about offloading. The results of their evaluation showed that the algorithm was able to perform as expected. proposed algorithm.

5.1 Overview of Optimization Techniques

Optimization is the ability of an act, which may be a formula, or model, in order in order to create something such as an arrangement or decision as flawless as it is feasible. It could also be described as "a mathematical discipline that concerns the finding of the extreme (minima and maxima) of numbers, functions, or systems" [13]. This refers to the minimization as well as maximization. The techniques of optimization can be employed to solve an optimization problem that could be single or multi-objective. Single-objective optimization is a method to tackle a single purpose to offer the best solution for each problem. Multi-objective Optimization helps accomplish multiple objectives at the same time and gives a range of possibilities instead of an all-encompassing solution [12]. The methods of optimization may be classified in linear as well as Nonlinear Programming Optimization. Programming optimization for linear programs is an objective

function in linear form which seeks to increase or minimize the impact of linear constraints. However nonlinear programming seeks to maximize or reduce objectives that are nonlinear [13]. In addition, nonlinear programming may be classified as deterministic or stochastic.
Deterministic optimization approaches employ mathematical models that are strictusing identical inputs, they can produce the same result each time. In this instance the best solution is typically local and is not able to be global. Therefore, deterministic optimization techniques are more appropriate for single-objective issues as opposed to multi-objective ones.

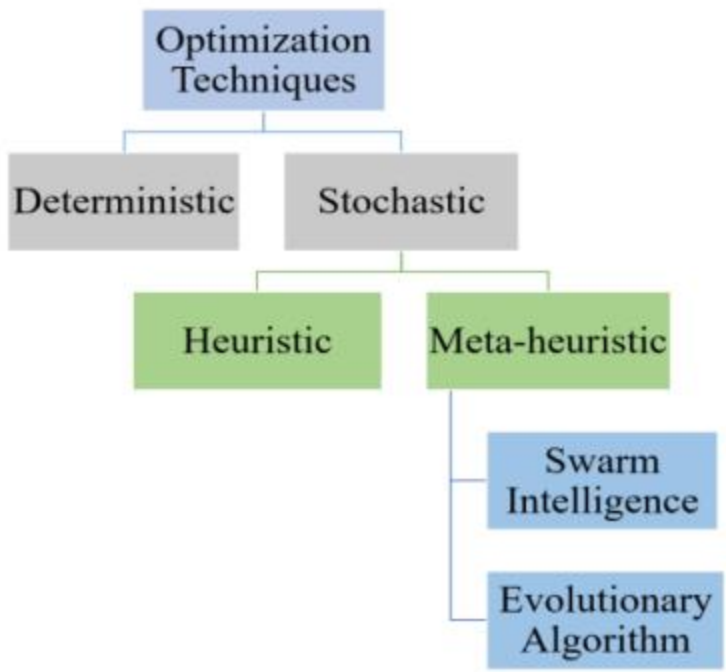

Fig. 3. Optimization techniques

5.2 Comparing Existing Solutions

The above solutions are summarized in Table 2, which shows that certain of the optimization strategies suggested to optimize offloading were predictable in nature, while others were based. AI-based systems typically depend on the software in the SI algorithm to optimize offloading. In actuality, SI supports heterogeneous, global and distributed environments , such as MEC. Additionally, SI is produced by automatic, self-organization and adaptive methods. In this regard, we examine the solutions more thoroughly in order to determine the best one to use in the study. In order to achieve this, the most appropriate requirements were determined as follows:

1.) Algorithm: SI algorithm is employed to maximize offloading.
2.) Distribution: Determines whether the architecture is centralized in cloud, or distributed on an edge layer.

3.) Optimization parameter: Parameters to be considered during the optimization.
(1) Time: The total execution time and transmission time for every task.
(2) Energy Consumption of energy for each job (execution as well as transmission).
(3) Cost the cost of the application's execution.
(4) Scalability The capacity of the application to expand without causing harm.
(5) Utilization of resources Utilizing scarce resources without creating an increase in the overhead of the system.
(6) Balance of load A capability to share work by collaborating with edge nodes which are under-loaded.
(7) Queue congestion is the avoidance of queue congestion in situations where the arrival rate of tasks exceeds the rate of service which results in system overhead.
(8) No. of repetitions: It is correlated to the duration of the work. In accordance with the requirements, there is a comparison of models based on SI.

6 DISCUSSION, OPEN ISSUES, AND FUTURE DIRECTIONS

It was demonstrated by this research that over the last few times, optimization methods are being developed to optimize offloading decision-making, especially Swarm Intelligence (SI) models because they provide a perfect match to the fluid and distributed component in edge computing. In reality, SI produces a better solution in the shortest time and is able to meet the requirements of real-time applications.

One of the major research areas is the integration of the SI-based offloading theories and edge computing to improve central solutions and make them more adaptable to the shared schema and while taking into account different objectives. As was mentioned previously, Cuckoo Search seems to be an ideal algorithm to tackle multi-objective offloading optimization since it delivers excellent performance in this domain when compared with others AI algorithm, which is shown in. In the end, applying Cuckoo into an edge-based system will enhance offloading and enable more secure solutions to assist mobile devices that run demanding applications. Furthermore, the offloading of computational computation to the edge could be enhanced by using CSA along with the parallel processing between edges that can boost the capacity of computation and reduces time..

7. CONCLUSION

The number of individuals using mobile devices to access the internet is rising. In fact, during the last several years, the number of individuals utilizing mobile devices has exploded. Mobile devices should be able to run real-time applications like gaming, e-commerce, healthcare, and many more. Additionally, users of mobile devices anticipate that the quality of service (QoS) they get will be of a high standard comparable to that of desktop-based programmers. On the other side, real-time applications need more resources, including the capacity to store data, power from the battery, computing power, and storage. Offloading involves delivering all or a portion of a mobile task to a high-end source, such as edge servers or cloud servers, where it is processed before being sent back to the mobile device. Making the most of a Smartphone's resources is made easier by doing this. Making the most of a Smartphone's resources involves offloading. On the other hand, offloading computation requires effort and time, both of which are crucial for real-time applications to operate at their best.

REFERENCES

[1] Alok Tripathi; Abhinav Mishra, "Cloud computing security considerations", IEEE 2011
[2] Yongmin Zhang; Xiaolong Lan; Ju Ren; Lin Cai,"Efficient Computing Resource Sharing for Mobile Edge-Cloud Computing Networks", IEEE, 2020
[3] Prasadu Peddi (2017) "Design of Simulators for Job Group Resource Allocation Scheduling In Grid and Cloud Computing Environments", ISSN: 2319- 8753 volume 6 issue 8 pp: 17805-17811.
[4] TAO Jing; JIN shen; TANG Jia; JI Yutong; ZHANG Ning,"Application of Cloud Edge Collaboration Architecture in Power IoT", IEEE 2020
[5] Tianchu Zhao; Sheng Zhou; Linqi Song; Zhiyuan Jiang; Xueying Guo; Zhisheng Niu"Energy-optimal and delay-bounded computation offloading in mobile edge computing with heterogeneous clouds", IEEE 2020
[6] Lih-Jen Kau; Cheng-Chang Chung; Ming-Sian Chen "A cloud computing-based image codec system for lossless compression of images on a Cortex-A8 platform",28-30 Nov. 2012, IEEE
[7] Nandiwardhana Waranugraha; Muhammad Suryanegara,"The Development of IoT-Smart Basket: Performance Comparison between Edge Computing and Cloud Computing System", IEEE, 2020
[8] Michael Fagan; Mohammad Maifi Hasan Khan; Bing Wang,"Leveraging Cloud Infrastructure for Troubleshooting Edge Computing Systems", 17-19 Dec. 2012, IEEE
[9] Per Skarin "Cloud-Assisted Model Predictive Control", IEEE, 2019
[10] Erdinc Korpeoglu; Cetin Sahin; Divyakant Agrawal; Amr El Abbadi; Takeo Hosomi; Yoshiki Seo,"Dragonfly: Cloud Assisted Peer-to-Peer Architecture for Multipoint Media Streaming Applications" , IEEE, 2013
[11] Prasadu Peddi (2020), MINING POSTS AND COMMENTS FROM ONLINE SOCIAL NETWORKS, Turkish Journal of Computer and Mathematics Education, Vol 11, No 3, pp: 1018-1030
[12] A. Akinbi; E. Pereira; C. Beaumont, "Evaluating security mechanisms implemented on public Platform-as-a-Service cloud environments case study: Windows Azure", IEEE, 2013
[13] Utsav Drolia; Rolando Martins; Jiaqi Tan; Ankit Chheda; Monil Sanghavi; Rajeev Gandhi; Priya Narasimhan"The Case for Mobile Edge-Clouds", IEEE, 2013
[14] Eyal De Lara; Carolina S. Gomes; Steve Langridge; S. Hossein Mortazavi; Meysam Roodi, "Poster Abstract: Hierarchical Serverless Computing for the Mobile Edge", IEEE 2016
[15] Prasadu Peddi (2016), Comparative study on cloud optimized resource and prediction using machine learning algorithm, ISSN: 2455-6300, volume 1, issue 3, pp: 88-94.

[16] Hongxing Li; Guochu Shou; Yihong Hu; Zhigang Guo"Mobile Edge Computing: Progress and Challenges", 2016, IEEE
[17] Yogesh Hole et al 2019 J. Phys.: Conf. Ser. 1362 012121

Author Bibliography:

Naga Lakshmi Somu is working as Assistant Professor in St.Ann's College for Women Hyderabad, Telangana, India. She is Research Scholar at Department of Computer Science & Engineering, Shri Jagdishprasad Jhabarmal Tibrewala University, Jhunjhunu, Rajasthan, India. She has obtained her MCA St.Ann's College for Women,Hyderabad , Telangana, India. Her research interests are Internet Cloud Computing.
Email: lakshmi.msccs@gmail.com