The demands of mobile applications and Internet of Things on cloud computing have given rise to the need to move cloud infrastructure closer to edge devices. Emerging architectures such as edge computing, fog computing and cloudlets aim at reducing data transfer times and removing the performance bottle neck of the traditional cloud. These new paradigms also enable advanced applications such as smart cities, smart buildings, and smart traffic light systems.

Introduction:

CloudIoT concept was introduced as the two individual technological advancements, Internet of Things and Cloud Computing complemented each other. The Internet of Things paradigm involves interconnection of intelligent devices and self-configuring nodes that can sense information and communicate without human interaction. These edge devices or nodes have limitations in terms of computational power and storage capacity and battery life. Cloud on the other hand has virtually unlimited computational power and storage capability, it provides interoperability for heterogenous IoT edge devices and is ubiquitous. CloudIoT and Cloud of Things aimed at integrating these two paradigms to overcome the limitations of edge devices and enable the IoT applications.

Smartphones and 3G/4G/LTE technologies has given rise to a plethora of mobile applications. Heavy compute-intensive speech recognition applications like OK Google and Siri or face recognition applications like snapchat are commonly used by a large number of users. Besides these, smart home applications or smart traffic management applications like Waze are also widespread.

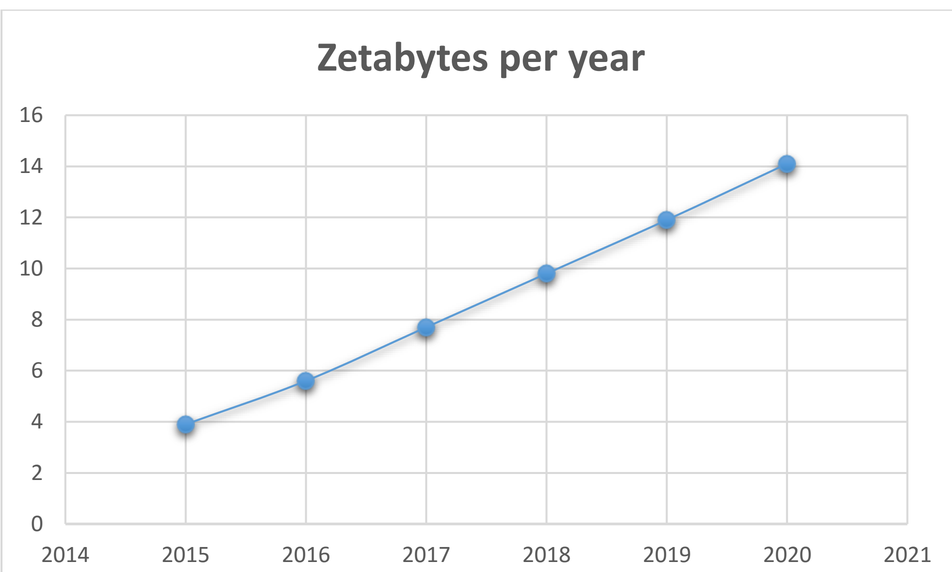

Figure 1: Number of "things" that will be connected to the internet by 2020. Source Cisco Systems

With the proliferation of IoT and mobile applications, the number of interconnected devices is increasing exponentially, Cisco Systems have predicated an estimated 50 billion devices to be connected to the internet by 2020. Data generated by these devices is also very large, for example the Cisco Global Cloud Index has estimated that "things" will produce 500 zetabytes of data but the IP generated cloud traffic will increase to 14.2 zetabytes by 2020. Traditional cloud computing has its limitations with regards to IoT applications that require low and reliable latency for its real-time critical applications. Mobile applications normally use WAN to communicate with the cloud, this normally has a low bandwidth. The cloud computing approach of making a round trip to the cloud will reach its performance bottleneck despite its data processing speeds and computational power, due to high data transfer rates and network bandwidth restrictions.

For example, external oil rigs generate 500GB of data weekly [1] or vehicles in a smart traffic management system can generate a few terabytes of data, if cloud computing is used for these applications it would end up increasing network traffic in that area and the response time would be very high.

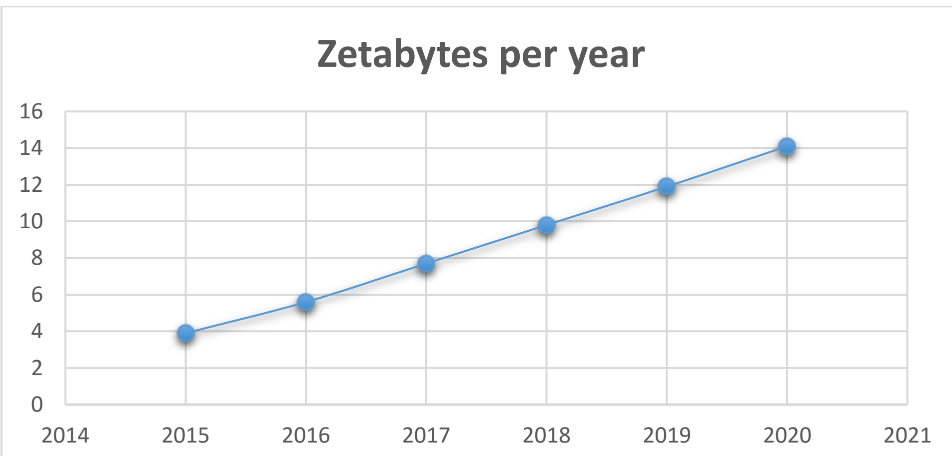

Figure 2: Traffic Growth Source : Cisco Global Index

The other limitations of traditional cloud computing for IoT applications and mobile applications involve, reliable connectivity (for fast mobile applications): loss of connection to the cloud can adversely affect critical, real-time mobile applications, geo-distribution (sensor networks, oil rig monitoring, pipeline monitoring applications) large -scale distribution (applications such as smart cities, smart grids).

The need to solve these issues of the cloud has resulted in the new paradigms, i.e. Fog/edge computing, and cloudlets. It is important to note that each of these paradigms have overlapping concepts and architectures. They all share the basic idea of moving computation and storage and other "cloud-like" features closer to the edge devices. This paper surveys these new architectures, highlights their advantages, challenges, and open research problems.

Section II

Fog Computing:

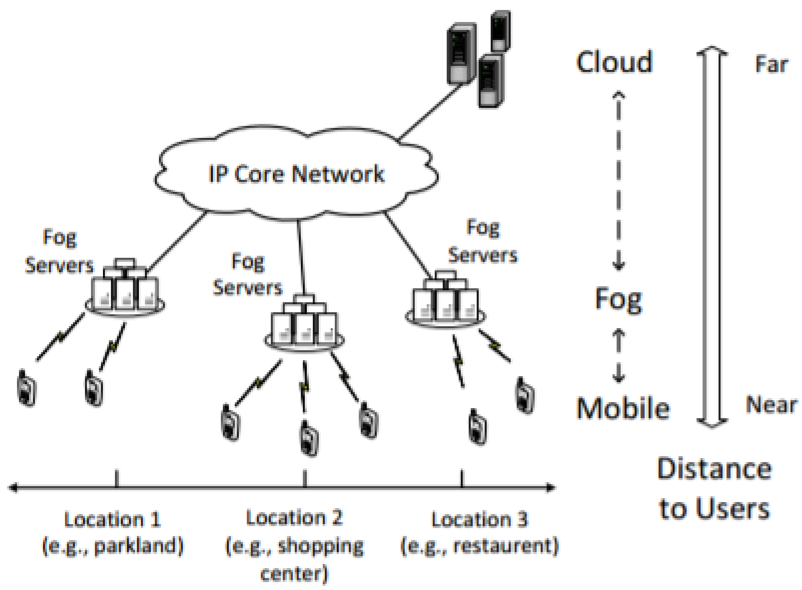

Figure 3: Fog Computing Architecture

The fog extends the cloud closer to the edge devices where the data is generated. This approach accelerates awareness and response to events by eliminating a round trip to the cloud for analysis. The fog architecture consists of an intermediate layer of fog nodes close to the edge devices. These fog nodes can be any component of the network connection. Fog nodes can range from resource poor devices such as sensors of light switches to more powerful components in the network such as routers or gateway switches. Fog computing is relevant for IoT applications that generate large amount of data require very low and predictable latency, geo-distributed applications (sensor networks, oil rig or pipeline monitoring), fast mobile applications (connected vehicles), large scale distributed control systems (smart grids, connected rail, smart traffic light systems)[2].

Related Work:

Fog computing was first introduced by Cisco Systems in 2012 as a highly-virtualized platform that provides compute, storage and networking services between end devices and the traditional cloud [3]. This paper described the use of fog computing in IoT applications such as connected vehicles with its advantages of geo-distribution, low reaction time, mobility and awareness. Fog computing benefits wireless sensor networks that comprise of resource poor elements. Fog nodes can enable a closed loop stable control systems for such applications. Fog computing does not cannibalize cloud computing but complements and it and while applications require localization some also need globalization of the cloud for analytics and big data. Fog computing is a hierarchal layered architecture where a higher tier means more geographical coverage and the cloud provides global coverage. [2]

Migration method for fog and cloud resources are described in [3]. It ensures application-defined end-to-end latency restrictions and reduces the network utilization by planning the migration ahead of time. They also show how the application knowledge of the complex event processing system can be used to reduce the required bandwidth of virtual machines during their migration. Network intensive operators are placed on distributed Fog devices while computationally intensive operators are in the Cloud. Migration costs are amortized by selecting migration targets that ensure a low expected network utilization for a sufficiently long time. This work does not optimize workload mobility because Fog devices are also able to carry computationally intensive tasks. It also does not optimize the size of control information or mobility overhead, and does not describe network control policies for finding optimal paths for different applications.

The context aware fog infrastructure was defined in [4] where the context of the different elements of the infrastructure can be extracted to create a virtualized network. The virtualized fog is managed by context manager and the SDN network manager. The SDN based network manager performs traffic engineering, load balancing and fog/edge computing based on the virtualized model. This evolvable context-aware architecture can be used in edge diverse IoT applications.

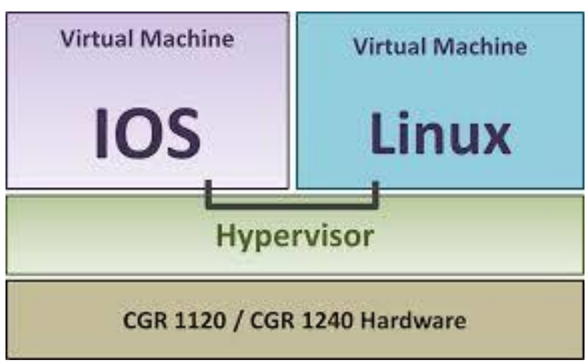

Figure 4: Cisco IOx framework for fog computing

Fog computing has evolved since its initial definition and now is a paradigm of itself with fog computing frameworks for IoT programmers. The Cisco IOx offers developers to create IoT applications and have them run on the network edge devices. The application is hosted on guest OS running in a hypervisor on Connected Grid Router. This runs the application on Cisco's powerful router making it a fog node thus implementing fog computing paradigm.

OpenFog consortium formed in November 2015, aims at an open architecture for ubiquitous fog computing for IoT applications. The OpenFog architecture aims at solving some of the open research issues in fog computing i.e. interoperability, scalability and security along with performance, reliability, availability and serviceability.

Cloudlets:

Mobile cloud computing has its own limitations in terms of latency and network bandwidth, with increasing traffic, reaching the "distant" cloud induces delays due to large data transfer times.

Cloudlet is an internet infrastructure element that arises from the convergence of mobile computing and cloud computing. It represents much like fog computing, the middle tier of the mobile device - cloudlet-cloud architecture. Cloudlets are soft state only, which means they may contain cached data from the cloud or buffered data from the mobile devices but no hard state. They are computationally powerful and resource rich and are well connected to the Cloud. Cloudlets need to be in physically close to mobile users and since these cloudlets are shared with various mobile devices security is a key feature.

Related Work:

This section discusses the related work on different cloudlet architectures.

A. VM Based Cloudlets:

The VM based cloudlet architecture is the one where mobile devices use the virtual machine technology to rapidly instantiate customized service software on a nearby cloudlet. [5]. These cloudlets are essentially a cluster of multicore computers with gigabit internal connectivity and a high bandwidth wireless LAN. Reduced latency along with short and predictable end-end response time on mobile applications is achieved due to the one hop wireless access to cloudlets. These cloudlets are easily manageable, they are soft state only and may contain cacheable data from the cloud of buffered data from mobile devices. The VM based approach is malleable since it can cater to a wide range of users with different devices and OS.

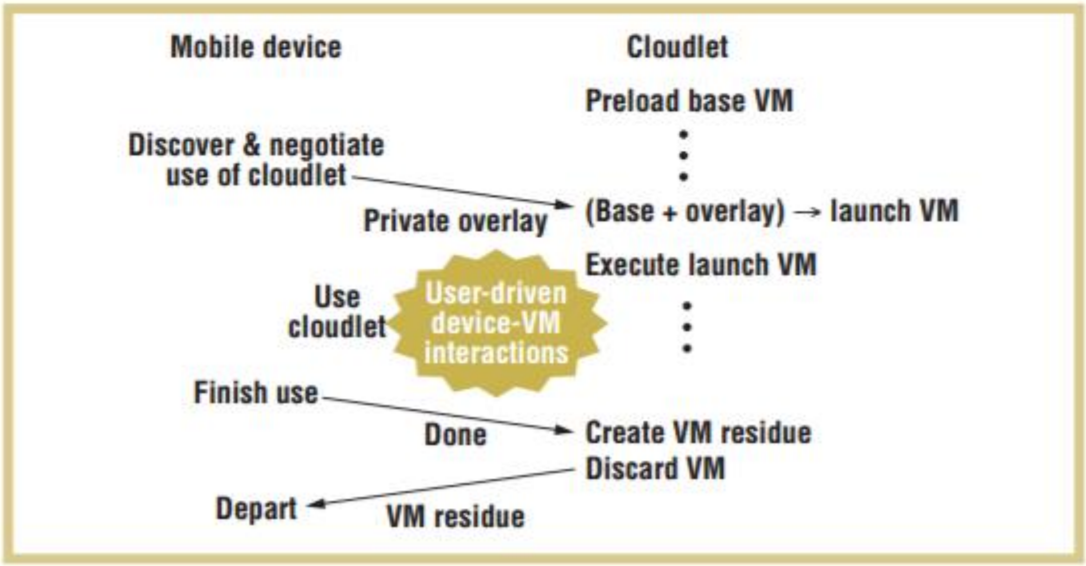

Figure 5: VM based Cloudlet Architecture

1. VM Migration model: In this model an application on the mobile device is stalled and encapsulated in a VM. This VM image is migrated on the cloudlets infrastructure where the application is unpacked and resumed. After execution on the cloudlet the VM state is saved and migrated back to the mobile device. This model thus requires the mobile device to have a VM image suspended in a resumable state.

2. VM overlay model: As described in the [6] a future with widespread use of cloudlets is foreseen, much like the wi-fi access points today. Thus, one of the main goals of this architecture is to make the deployment and management of cloudlet infrastructure trivial or easy. At the same time not tightly restricting the software to the cloudlets to allow support for myriad applications. This is achieved by the transient customization of cloudlet infrastructure using hardware VM technology. The goal of this architecture is to provide a VM that encapsulates and separates the guest software from the cloudlet's host software. The pre-customization and post-clean up tasks ensure that the host software state is achieved after every use. The VM overlay model consists of an overlay VM sent from the mobile device to the cloudlet that already contains the base VM. The modified VM is launched and the application is executed. An example of this could be a speech recognition application, a base VM receives captured speech from a mobile device which it can process and return the output to the mobile device.

   In this model the VM overlay on the mobile device has to be up-to date with the base VM on the cloudlets. Another important assumption of this model is that a relatively small number of VMs will accommodate a wide range of applications. Further research in this model needs to be done to minimize latency involved in overlay transmission time and decompressing and compressing the VMs for real-time deployment of this model.

3. On-demand VM Provisioning Model: In this model the attempt is to reduce the run-time transfer cost that is seen in (2). A mobile device discovers a cloudlet and transfers a provisioning script to it. The cloudlet creates an on demand VM that matches the requirements of the mobile device using this script. If the cloudlet already has the required VM in its repository, then it can instantiate that VM to run the application.
   This paper [8] proposes intelligent systems for VM provisioning to meet performance goals of the mobile devices. In the proposed system, they use the NIMO system to build application performance models and use a Java toolkit, SHIRAKO for leasing the utility resources.

   It is important to note that for a thin-client (a device with low computation power) mobile users performance degradation will be observed in each of these clients when the users are in transit. This is because physical proximity to the cloudlets is key to reduced latency and fast response times. When these mobile devices are “out of range” they may be rendered to using the cloud, or local resources on the device for compute-intensive applications which will induce latency. Paper [7] introduces live migration techniques for VM Handoff to “handoff” execution from one cloudlet to another to sustain performance for a mobile user.

B. Mobile Cloud Hybrid Architecture (MOCHA) :

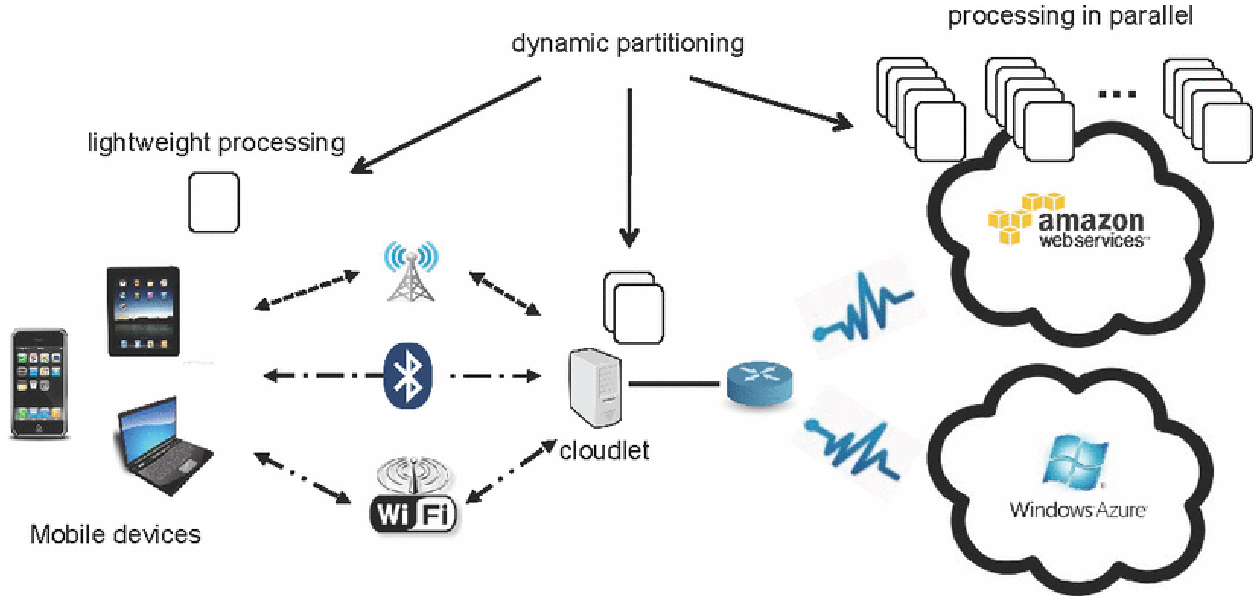

Figure 6: Mobile Cloud Hybrid Architecture.

The Mobile Cloud Hybrid Architecture uses cloudlets as an intermediate layer between the cloud servers and mobile devices for compute-intensive applications. The MOCHA was created to target battlefield mobile applications that require massively-parallelizable cloud servers. Here the cloudlets decide the partitioning of the offloading among the cloudlet and the cloud servers to optimize the overall performance. The mobile devices in this architecture are used to capture data and send it to the cloudlet for pre-processing. The mobile devices may be connected to each other and the cloudlet through lightweight wireless protocols, the cloudlet in turn is connected to the cloud server using Internet or satellite connection.

This architecture can be useful for real-time critical applications that use wireless sensor networks like battlefield object detection, airport security, enhanced AMBER Alert.

[9] implemented a face recognition for airport security system using the MOCHA architecture and observed better performance when using an intermediate layer of cloudlets since it reduced the data transfer time and latency. It was also observed that a more intelligent partitioning algorithm had a positive impact on the performance.

C. MicroData Centers (By Microsoft):

Microsoft's hyper scale data centers placed globally that carries 110 million petabytes of data over 1.4 million miles of routes. These hyper scale data centers provide Azure cloud services globally and aim at high performance, lower latency and lower transactional cost. According to Victor Bahl, even after these datacenters the increase in traffic growth stresses the underlying cores and increases latency and for reliable high-performance data transfer 4 times the existing optical fiber may be needed. According to him the solution to reduce latency further is not building more hyper scale data centers but micro data centers closer to the edge users.

Building an extensive infrastructure of microDatacenters (mDcs) , i.e. 1-10s of servers with several TBs of storage placed everywhere. These micro Datacenters are classic Content Delivery Network (CDN) nodes that can improve the performance of search engines, office productivity tools and other cloud services. The benefits of these microDatacenters include reduced latency, better reliability due to diverse paths, lower device cost for high end gaming among others. These mDcs can also be connected to edge devices such as fog nodes for integration of these technologies.

Although fog computing and cloudlets might be different paradigms they have overlapping concepts. High performance, low latency requirements and network bandwidth restrictions drives both these paradigms. Given similarities in concepts and motivation for these paradigms they have similar challenges and open issues that are discussed in the next section.

Challenges and Open Research Problems:

1. Security and Privacy Threat:
   Security is a key feature in fog computing since the fog nodes and cloudlets are distributed systems and spread over different components of the network infrastructure. Multiple users sharing resources over a distributed system gives rise to security and privacy threats.
   Even though the VM based cloudlets are more secure because of the thick VM insulating it a malicious VMM can distort applications without the user knowing of it. [6]. Man in the middle attack can adversely affect fog computing, in this a fog gateway node may be compromised or replaced by a mock device. Another challenge is access control of data in fog computing and cloud computing, it is important to have access control techniques that can have fine-grained access over the client/fog node -cloudlet-cloud architecture. Some of the linux kernel schemes discussed in[15] such as Simplified Mandatory Access Control SMACK and Discretionary Access Control (DAC) are possible solutions to this problem.

   Data privacy along with location privacy are other challenges for fog computing and cloudlets. Data leak in fog computing for smart grids and smart home can reveal a lot of information about a user and track their actions. E.g. when they are more likely to be at home etc. Also, information about which fog nodes (servers, gateway devices) are being used puts location privacy at risk. Using multiple fog nodes is a possible solution, however the discovering the location by tracking trajectory of the network makes this solution ineffective.

2. Interoperability:

Since fog nodes consist of hierarchal architecture of heterogenous components involving sensors, smart meters to gateway devices and servers it is important to establish communication between these different devices. Interoperability is key to reliable and secure communication between edge devices. This requires standardization and naming schemes for edge-centric computing. The Open Fog Consortia and OPC foundation are working towards, specifications that standardize communication of data to multi-vendor enterprise systems and between production devices. The use of Software Defined Network to manage the network architecture of fog nodes is a solution to interoperability.

Since there will be different components involved in the cloudlet and fog architecture these components can be owned by different service providers and it is important to standardize and enable interoperability between them for the deployment of these technologies in the real world.

3. Programmability:
   Since the application functions are distributed from the cloud to the edge devices it is important to employ a method that can synchronize updates for the different functional units of the applications. Easy to use frameworks and tools can provide a scalable and extensible solution to this issue.

4. Scalability:
   Scalability is an important feature for fog computing architectures and cloudlets. These infrastructures must be scalable. As the number of IoT applications or mobile applications increase the fog network and cloudlets should be able to grow and accommodate these changes. These paradigms must also be scalable in terms of performance and with increased applications the performance should not become a bottleneck.

Conclusion:

Fog/edge computing and cloudlets are essential for advancement of IoT and mobile applications. These paradigms reduce response times, latency and jitter from the traditional cloud and bring advantages of location awareness, geo-distribution and reliability thus enabling advanced applications like smart building management and smart grids. Neither of these paradigms replace the traditional cloud instead they form an intermediate layer between the cloud and the mobile/edge devices for optimum performance. Security is a major concern for these paradigms and open research problems include, computational offloading, scalability and interoperability. Future work on this paper would explore the synergy between fog/edge computing and cloudlets and the possibility of convergence or integration of the two approaches for advanced applications.

References:

[1] Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are , White Paper from Cisco

[2] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan and Jiang Zhu, Fog Computing: A Platform for Internet of Things and Analytics

[3] Ivan Stojmenovic, Sheng Wen, The Fog Computing Paradigm: Scenarios and Security Issues, 2014

[4] Won Sang Chin, Hyun-soo Kim, Young Ju Heo, Ju Wook Jang, A Context-based Future Network Infrastructure for IoT Services

[5] A.S. Jaiswal, V.M. Thakare, S.S. Sherekar, Performance based Analysis of Cloudlet Architectures in Mobile Cloud Computing

[6] Mahadev Satyanarayanan, Victor Bahl, Ramón Cáceres, Nigel Davies, The Case for VM-Based Cloudlets in Mobile Computing

[7] Kiryong Ha, Yoshihisa Abe, Zhuo Chen, Wenlu Hu, Brandon Amos, Padmanabhan Pillai† , Mahadev Satyanarayanan, Adaptive VM Handoff

[8] Piyush Shivam, Azbayar Demberel, Pradeep Gunda, Automated and On-Demand Provisioning of Virtual Machines for Database Applications

[9] Tolga Soyata, Rajani Muraleedharan, Colin Funai, Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture

[10] Usman Shaukata, , Ejaz Ahmedb, , Zahid Anwara, , Feng Xiac, Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges

[11] Rome Eugen Borcoci, Fog Computing, Mobile Edge Computing, Cloudlets - which one? SoftNet 2016 Conference August 21, 2016

[12] Tolga Soyataa , Rajani Muraleedharana , Jonathan Langdonb , Colin Funaia , Scott Amesc , Minseok Kwond , Wendi Heinzelmana, COMBAT: mobile-Cloud-based cOmpute/coMmunications infrastructure for BATtlefield applications

[13] OpenFog Architecture, White Paper.

[14] Victor Bahl , emergence of micro datacenter (cloudlets/edges) for mobile computing , Microsoft,May 13, 2015

[15] About Securit solutions in Fog Computing, Eugan Petac, Andreea-Oana Petac