Overview - Fog Computing and Internet-of-Things (IoT)

C. S. R. Prabhu

1. INTRODUCTION

1.1 A New Economy based on IoT Emerging By 2025

The internet today is getting connected to a very large number of devices or sensors of IoT. These numbers were not envisaged previously. IoT or Internet of Things, alternatively known as IOE, Internet of Everything envisages connecting more and more consumer electronic devices, home appliances, medical devices, cameras, all types of sensors for temperature, pressure or humidity, etc., in addition to mobile phones and industrial IoT devices. The numbers are staggering: up to 50 billion devices are expected to be connected according to Ericsson. This will result in a 20 trillion market opportunity in less than 10 years. In 2025 it is estimated that IoT will have an economic impact with revenue generated and operational ravings up to 11 trillion dollars per year on 11 percent of world economy with common public users deploying upto 1 trillion devices 2025 i.e., the technological changes of an IoT drives economy.

1.2 The Technological Challenges of an IoT Driven Economy

Firstly the IoT devices are characterized by limited storage and limited computational power. This resulted is very large number of cloud services coming up, to provide the necessary support to these smart devices.

This resulted in over loading of the present networks with inadequate bandwidth. The joint adoption of Network Function Virtualization (NFV) and Software Defined Networking (SDN) architectures was considered to be a promising solution.

The cloud paradigm of Infrastructure as a Service (IaaS) was capable of providing virtual computing resources to remote users also, over the Internet.

However, the performance of such cloud services was constrained by the performance loads offered by the respective Servers in Data Centres which host these cloud services. Such data centres are usually located at a close proximity to core networks which are at large distances from the edge or field locations where IoT devices or sensors are located for gathering the data.

This results in the end-to-end latency to become two orders of magnitude higher than the required values for delivering the expected performance of cloud services

The problem is further compounded by the fact that the data generated by the IoT devices is an unprecedentedly large volume is real time and could be of great value, if analysed in real time.

However, the data volumes generated overwhelm the storage systems and analytical applications. Cloud services (such as IaaS) could have helped by offering on-demand and scalable storage and processing services that can scale up to IoT requirements.

However, as indicated above, in real life situations such as health monitoring, emergency response and management and other latency sensitive applications., the delay caused by transferring this large data to the cloud and getting back the results to the application is not realistic or satisfactory and therefore not acceptable.

It is also not advisable to send the large data to the cloud, as it will saturate and overload the whole network bandwidth and affect its performance, as a whole, affecting all other o IoT applications badly.

In real life situations of current trends such as real time health care based an IOT applications, Smart Cities, Smart Villages, it has been noticed that the overall network traffic will be affected adversely.

1.3 Fog Computing Paradigm as a Solution

In order to affectively address the above mentioned challenges, initially Edge computing paradigm was proposed to use its computing resources for the purposes of local storage, preliminary data processing so that network load or congestion could be reduced and also enable localized data analysis so as to result in fast data driven decision making process. Unfortunately, however, the edge computing devices had very limited resources which will result in resource contention and increases processing latency.

Therefore, a near paradigm known as Fog Computing was emerging, which seamlessly integrates the edge devices on one hand with the cloud devices on the other, to overcome all the limitations of Edge computing as well.

Fog computing, therefore, avoids resource contention at the edge by leveraging cloud resources and also coordinating geographically distributed edge devices.

1.4 Definitions of Fog Computing

Fog computing has been defined in many ways:

1. Fog computing is an extension of the cloud platform from the Core to the Edge of the network (Bonomi et al., 2012; Dastjerdi et al., 2016).
   --this definition is too simplistic ,without noticing the key aspects of ubiquitous nature of Fog, its enhanced networking capabilities such as providing hosting environment and improved support for interaction between devices.
2. A different autonomous definitions of Fog by Rodero-Morcho:
   Fog computing is a scenario where a large number of heterogeneous (wireless and also autonomous) ubiquitous and decentralized devices communicate and cooperate among themselves and also with the network to perform storage and processing tasks, without a third party interaction for supporting basic network functions or for supporting new services and applications that run in a sandboxed environment with users leasing part of their devices to host these services and get incentives for doing so.
   The above definition fails to address the crucial connections of fog to the cloud.
   Then we have a third definition by S. Yi et al as:
3. Fog computing is a geographically distributed computing architecture with a resource pool consisting of one or more ubiquitously connected heterogeneous devices (including edge devices) at the edge of network and not exclusively seamlessly backed by cloud services, to collaboratively provide elastic computation, storage and other services and tasks in isolated environments to a large pool of clients in proximity.
   We can expect more new definitions In future.

1.5 Characteristics of Fog Computing

While elastic resources of computation and storage and networking are common for both Cloud and Fog, we can identify the unique characteristic of Fog computing paradigm which make it a non-trivial extension of cloud as follows:

1. **Location at the edge:** Being located close to the edge, fog has the capability to support latency-sensitive applications that require real-time processing.
2. **Location awareness:** Unlike cloud services being widely distributed in nature, the services offered by fog are required to possess the ability to derive their locations and also track the. End user's device location.
3. **Real-time interactions and service delivery**: Unlike cloud based bulk processing, fog applications ensure real-time service delivery which is not feasible in cloud services due to latency and overloading of the network.
4. **Edge analytics**: As against centralized analytics, fog computing can support analysing sensitive data locally itself (instead of sending the data all the way to the cloud for analysis).
5. **Scalability:** Cloud may not be able to handle all data is real time. Fog computing alleviates this problem of the cloud becoming overloaded and not being able to deliver real time analytics. Not all analytics is essentially required to happen at the cloud (which calls for all data from all the edges to reach the cloud). It is often more appropriate and adequate to perform the pre-processing at the edge itself and send only the most essential and relevant information to the cloud.

To summarize, fog computing is a distributed computing paradigm that provides cloud like services to the edge of the network.

Fog computing leverages a) cloud and b) edge resources along with its own resources.

In essence, fog computing technology deals with IoT data locally by utilizing clients or edge devices near to the end users to carry out a substantial amount of storage, communication, central, configuration and management. This intermediation approach of the fog benefit from its close proximity to the edge devices such as sensors, while at the same time, leverages the on-demand scalability available in the offerings of cloud services(as IaaS).

Fog computing also involves the components of data processing or analytics applications running in distributed cloud and edge devices. Therefore, fog computing facilitates the management and programming of the computing, networking and storage services between data centres on one side and end or edge devices on the other. In addition, by its own vary nature, fog computing supports user mobility, resource and interface heterogeneity and alsodistributed data analytics for meeting the requirements and expectations of widely distributed edge applications which calls for real time response with very low latency.

1.6 Architectures of Fog Computing

Cloudlet, IoX and Paradrop were early fog computing architectures (through fog computing did not emerge by then fully)

1.6.1 *Cloudlet Architecture (Satyanarayana, et al., 2015)*

Cloudlet, as shown below, is anexample of implementation of a resource rich fog mode

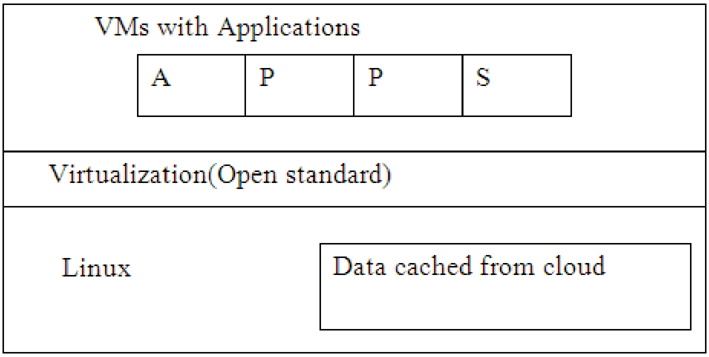

Fig. 1 Cloudlet

In a three layer design the bottom most layer is Linux Operating System and a data cache from cloud. The virtualization layer is middle layer with standard IaaS tools as open stack. The top layer is VM instancessupportingvarious applications.

1.6.2 *IoX Architecture*

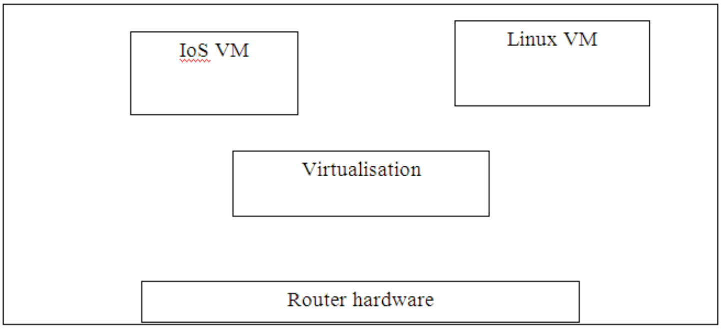

Fig. 2 IoX Architecture

IoX is a CISCO router based architecture. Applications are hosted in guest OS running on hypervisor upon the hardware of a grid router. The platform supports developments of scripts, code, and install own operating system. This platform is not open to public and relies onexpensive hardware.

1.6.3 *Local Grid's Fog Computing Platform*

It is an embedded software or installed on network devices and sensors. It standardises and secures communication between all kinds of devices across all vendors, then minimizing customization and services costs. Local Grid's platform resides on devices between the edge and the cloud and provides reliable M2M communication between devices without having to go through the cloud. This allows applications to make real time decisions right at the edge without having to deal with the high latency of communicating with the cloud. Moreover, all the local grid devices can communicate with the cloud through open communication standards, realising the concept of fog being an extension of the cloud.

Applications running on Local Grids platform can utilize the interplay between the fog and the cloud to solve more complex problems. Local Grid's fog computing platform is packaged together and shipped with Local grid "vRTU", a software based virtual remote terminal unit that transforms communications between edge devices into compatible open standards. vRTU can be installed on the off the shelf as well as custom solutions from OEMs, endowing devices with RTU capabilities and providing of single point for management of all the edge devices, thus cutting down customization and maintenance cost. .

1.6.4 *Parstream*

Parstream is a real time IOT analytics platform. CISCO joined hands with Parstream to build a fast, reliable and highly scalable infrastructure for analysis on the fog. Parstream offerings of Big Data analytics platform for IoT is contingent on its patented data base technology, Parstream DB, which is a column based, In Memory data base with highly parallel and fault tolerant architecture built using patented indexing and compression algorithms. Being In Memory, it is ideally suitable for fog devices with limited disk space. Any query can be executed at the edge and analytics be performed in a distributed manner. It can be deployed in fog enabled devices such as Cisco IoX.

1.6.5 *Para Drop*

It is implemented on a gateway (Wi-Fi access point or home set top box) which is an ideal fog node being very close to the end user. Para Drop is designed from home usage scenarios only and is also used as a fully decentralized manner where all application servers are required to use a Para Drop server as entry points to services provided by gateways. Then we can conclude that Para Drop is a meant for light weight task scenarios.

1.6.6 *Prismatic Vortex*

It is a ubiquitous data sharing platform made for internet of things. It provides scalable end to end seamless, efficient secure at timely data sharing access Iota supporting edges, gateways and the cloud.

VORTEX leverages the DDS 2.0 standard for interoperable data sharing and extends it to supports internet scale, mobility and web 2.0 applications. VORTE seamlessly interacts with common IoT message passing protocols as MQTT and COAP.

VORTEX also provides fine grain access control support and both symmetric and asymmetric authentications. Each IoT device is connected to a vortex edge device that executes all vortex software each platform of which software performs a function necessary for the realization of globally shared DDS.

A vortex edge device with IOT devices connected to it forms a domain (a DDS entity) called fog domain in this context.

Equipped with such devices VORTEX supports a number of deployment models:

1. ***Fog + cloud:*** IoT devices inside a fog domain communicate with each other in a peer to peer fashion. Between such two or many fog domains they have to communicate through the cloud.

2. ***Fog+cloud-link+cloud:*** Similar to the previous model, devices with in the same fog domain communicate peer to peer while devices not in the same fog domain exchange data through the cloud using a cloud link that handles the associated security issues and controls what information is exposed.
3. ***Federated fog****:* Each fog domain has a vortex cloud link running on the vortex device. Federated fog is a collection of fog domains which are federated by cloud link instances which control the information exchange between fog domains.

1.7 Designing a Robust Fog Computing Platform

The design goods for a robust fog computing platform are:

1. **Latency**: All fog applications should have low latency orexecution time of a task, the task offloading time, time for cyber foraging and speed of decision working.
2. **Efficiency**: Efficiency is utilizing resources and energy is very essential and more necessary than in the case of cloud scenario close to a) not all fog nodes are rich in resources than have very limited computational power measuring and storage b) fog odes and clients are battery power operated such as hand hold devices, wearable devices and wireless sensor units.
3. **Generalization**: As fog nodes and clients may be very heterogeneous, we need to provide the same abstraction to top layer applications and services for fog clients. The API should be general to cope with existing protocols of API. (M2M protocols, smart appliance API's, and smart vehicle API's).

1.8 Present Challenges in Designing fog Computing Platform

The following challenges can be identified in designing fog nodes:

a) ***Choice of virtualization technology:*** this will determine the efficiency speed and flexibility available in fog nodes.

b) ***Latency:*** Latency needs to be minimized as fog applications are necessarily expecting real time responses.

The following considerationsare important for bringingdownlatency

(i) *Data aggregation:* If data aggregation is not completed before data processing it leads to delays. By applying techniques such as data partitioning/filtering and utilizing locality in hierarchy we can reduce data volumes for computation at higher layers.

(ii) *Resource provisioning:* If resource schedulingand provisioning is not performed in proper time for resource constrained fog nodes it will lead to latency. This calls for correct scheduling using priority and mobility model.

(iii) *Node mobility, churn or failure:* in the event of node mobility, churn or failure of nodes, fog computing will be affected. Mitigation strategy such as chuck pointing, rescheduling and replication can be deployed.

c) ***Network management:*** For fog functionality network management is critical. Deploying software Defined networking (SDN) or network function virtualization (NFV) integrated together into fog computing is going to be challenging.

d) ***Security and privacy:*** Access control and intrusion detection system is to be deployed with due support from every layer of the platform.

1.9 Platform and Applications

1.9.1 *Components of Fog Computing Platform*

The components of a general fog computing platform can be identified in the following manner:

1) **Authentication and authorization**

   Every user who attempts entry into a fog computing environment needs to be authenticated and authorized with appropriate novel schemes. This will help in identifying the user access pattern,mobility pattern and trusted security devices.

   Access control scheme for authorization of heterogeneous resources is possible as proposed in C. Dsozaya,G-J aho and M.Tejaimod" Policy driven security management for Fog computing: preliminary framework and case study" In IEEE international conference on information reuse and integration,2014).

2) **Offloading management**

   Offloading in fog needs to solve several problems:

   a) What kinds of information are needed in offloading decision?

   b) How to partition applications for offloading?

   c) How to design optimal offloading schemes?

3) **Location services**: For delivering location services, we need to maintain a location list of neighbor nodes, both mobile and non-mobile, track mobile end users and also share location information among the involved fog nodes. Mapping network locations with physical locations, mobility model provided by end users can be adapted (or even mobility model learning is possible).

   The tracking and mapping on mobile nodes will need information from multiple layers (physical (ultra sound), wireless, signal and signature, GPS, IMCE sensor), network (IP) address and applications (Social activities), all of which call for new design for the component.

4) **System monitor**: In a fog computing platform as also in cloud infrastructure, a System Monitor becomes essential for providing crucial information such as work load, usage, energy, and other parameters to help a lot of decision making and also pricing.

5) **Resource management**: Resource management comprises tasks related to resource discovery, resource allocation, the dynamic joining and leaving of fog nodes, provisioning and maintaining the resource pool in a distributed manner.

6) **VM scheduling**: New VM scheduling strategies are required to be designed for finding optimal solutions due to fused inputs of system usage, workload statistics location information and mobility model.

1.9.2 *Applications and Case Studies*

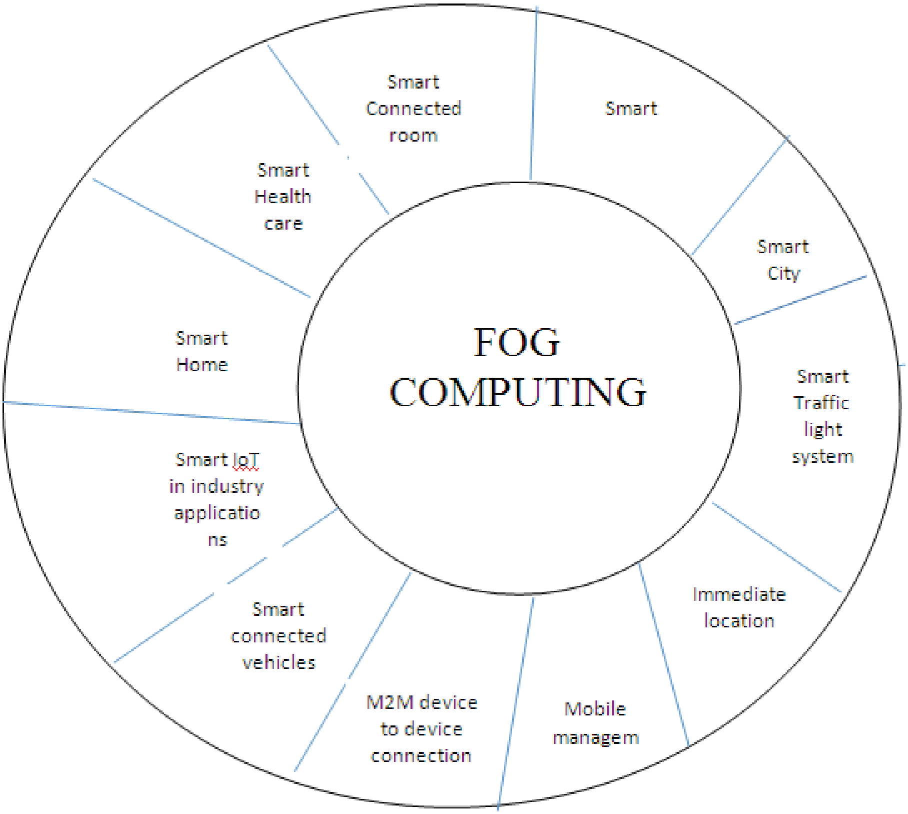

Fig. 3 Fog Applications

1.9.2.1 ***Health data management and health care:*** Health data management is a sensitive matter as privacy will be required for protected for personal health data. Fog computing helps the patients to take care of their own health data locally. Those health data will be stored in a fog nodes as a smart phone or smart vehicle. The computation will be out sourced in a privacy preserving manner whenever the patient is seeking help from a medical lab or a physician's office. Modification of data will happen directly in the fog node (Such as a smart mobile phone or a smart vehicle).

Cao et al.,(1) developed fast, fog computing assisted distributed analytics system to monitor fall for stroke patients.

A real time fall detection system based on fog computing that divides the fall detection task between edge devices and the cloud is developed. A full set of fall detection algorithms including acceleration measurements and time series analysis method and also filtering techniques to facilitate fall detection process are deployed.

In real world data conditions this system achieves a high sensitivity and specificity with efficient response time and energy consumption.

Stanchev et al., (2) proposed a 3 tier architecture for smart health care comprising of a role model, layered cloud architecture a fog computing layer to provide an efficient architecture for health care and elderly care applications. The fog layer improves the architecture by providing low latency, mobility support, and location awareness and also security. Using BPMN (Business Process Modelling Notation) and SOA (Software Oriented Architecture), services are identified with smart devices in a fog environment. A smart sensor based health care infrastructure is shown as a use case.

1.9.2.2 ***Smart village health care:*** In village a Primary Health Centre (PHC) alone is set up where a Doctor and support staff, diagnostic centre etc. Operated by the state government, where the basic diagnostics and clinical infrastructure is available. In smart village the facilities in a primary health centre are enhanced using smart health devices such as smart ECG, where a IoT enabled ECG is connected to a local fog server to upload the ECGs delivered in a day (up to 50).

Subsequently the fog server will upload the weekly data to the cloud. At any time each such ECG uploaded can be accessed and examined by a cardiologist if referred by the local doctor, if felt necessary. This results in Tele medicine with due interventions for speciality and super speciality medical experts all over the globe.

1.9.2.3 ***Smart home:*** Smart home is a home where many IoT functionalities can be deployed. This means many different smart devices and sensors are connected at home. However products from different vendors are difficult to work together. Some tasks which require large amount of computation and storage (for example real time video analytics) are infusible due to the limited capability of the hardware. To solve these problems fog computing is utilized to merger integrate all devices into a single platform and empower the identical smart home applications with elastic resources from the cloud.

Let us consider smart home security applications: widely deployed secure sensors consist of smart lock, video/audio recorder, sensor monitors for sensors such as light sensors occupancy sensors and motion sensors etc.

Unless all these sensors are manufactured by the same vendor, it becomes very difficult to integrate their functionalities. Therefore fog computing steps in. .

Fog computing can provide:

(a) Unified interface to integrate all kinds of independent devices.

(b) Flexible resources to support computation and storage by accessing the cloud resources as VMs and storage as necessary for provisioning.

(c) Realtime processing and low latency response.

We need to set up fog platform first and then we need to connect each sensor as a client. The corresponding sever application can be installed in independent VMs. The processing logic can also be implemented in the VM, so as to process the data shared by those secure monitor applications. For example, if a motion sensor detects a suspicious motion in a room, then a cleaning robot with a video camera will be given instructions to identify exact location.

Real time video analysis will process these video outputs and confirm whether the intruder is really there or whether it is a false alarm.

A notification or report will besent to the owner of the smart home and police will be altered if found necessary.

1.9.2.4 ***Smart vehicle and vehicular fog computing:*** Vehicular fog computing deploys smart vehicles has many possible applications such as

(a) Smart traffic monitoring and smart traffic light scheduling in city traffic for congestion mitigation and precaution sharing gas part of smart traffic management by traffic police.

(b) Parking Facility management by monitoring occupancy of parking lots and providing guidance information to car drivers regarding the availability of parking slots.

(c) Infrastructure based vehicular fog computing such as Tube (Luo, et al., 2011), an autonomous vehicular content distribution, which relies on fog nodes deployed along with road side: fog nodes are responsible to send/retrieve information to and from the vehicles being driven on nearby roads.

Autonomous vehicular clouds (Elto Weiss 2010 et al.,) utilize vehicles on the fly to form a fog and/or a cloud to support adhoc events: each fog can communicate with its own client or clients in other fogs.

1.9.2.5 ***Augmented reality applications:*** Augmented Reality applications necessarily demand very low latency as otherwise the user experience will be disturbed and damaged. Therefore fog computing has a great potential for augmented reality applications.Zoo et al., 2014built an augmented brain Computer interaction game based on fog computing and Linked Data. When a person plays a game EEG signals from brain are captured (using his/her EEG sensors analysed), classified into classes which can help identify the brain state of the player.

Brain state signal processing and classification is a heavy signal processing task requiring very high computational load of processing, to be carried out in real time, which is a very difficult task.

By employing a combination of both fog and cloud servers, it is possible to perform continuous real time brain state classification at the fog servers while the classification models are tuned regularly in the cloud servers based on the Degrading collected by the sensors.

Ha et al., proposed wearable cognitude architecture system based on fog Google glass devices to assist people with mental acuity. The cognitude devices deployed have constrained resources and therefore the computing intensive workloads of this application need to be off loaded to an external server. The response of this off loads should be real time responses without which the user experience will be a failure.

Offloading computing intensive takes to the cloud results in long latency and therefore nearby devices is put to use. These devices may communicate with the cloud for delay tolerant jobs like error reporting and logging. This scenario is a typical fog computing scenario in that the latency critical analysis at the very edge- in fog- and the latency tolerant computation at the cloud thus portraying fog as an extension of the cloud.

2. FOG APPLICATION MANAGEMENT

2.1 Introduction

Typically the IOT and Fog applications comprise of the IOT devices which carry out some common activities such as receiving data from the IOT devices, pre-processing the received data, performing the local analysis of the received data by using Fog analytics tools (residing on Fog Servers), handling of events of interest for responding back in real time, to the IOT application demands (1).

2.2 Application Management Approaches

Unlike cloud, the Fog nodes are very limited in resources with constraints, they are heterogeneous and distributed.

Therefore, to develop and deploy large scale applications in Fog environment, the applications are required to be modelled as a collection of light weight, interdependent Application Modules (2).

Each application module basically contains necessary instructions to execute the typical functions of an IOT application; receiving data, processing the received data, (from sensors or IOT devices) and performing analytics and / or give a response in real time back to the application concerned.

Any application module, therefore, contains instructions necessary for executing each of the above mentioned steps or components, each generating a corresponding specific output. Later based on data dependency, the output is sent to another module as input. For the purpose of execution, each module requires certain amount of resources such as CPU, Memory, bandwidth, etc. Thus, Application Modules together with their allocated resources constitute the data processing elements of different applications. For the purposes of distributed development of large scale applications, the decomposition of applications as above will be efficient and effective.

In order to reduce overheads on centralized cloud, distributed application development strategies have been discussed (Vaqueno and Roders – Moreno 2014) model programming platform (Hang, et al 203), and about coordination of Fog nodes during application execution (3).

2.3 Performance

What are the performance related issues in applications in Fog eco system? Latency related issues such as node to node communication, application service delivery deadlines, data receiving frequencies of different applications all are issues. These parameters influence QOS (Quality of Service), QOE (Quality of Experience), resource utilization and energy consumption of the Fog Applications.

2.4 Latency Aware Application Management

Various strategies for efficient application management in latency sensitive applications before the emergence of Fog Computing have been discussed for cloud or mobile cloud applications (4, 5, 6, and 7)

After the emergence of Fog computing, latency aware Application Module Management Policy for Fog environment is also being evolved that considers both deadlines and input receiving frequency of various applications with an objective to deadline driven QOS provisioning of applications and resource optimization for maintaining energy optimization in Fog. (8)

2.5 Distributed Application Development in Fog

Different approaches exist for Fog application management 1. "Droplets" – smallest parts of application to be placed in a distributed manner as Fog nodes as per requirements.

When billions of geo distributed IOT devices connected to a cloud start generating a very large number of service requests, Fog computing paradigm becomes compulsory supplement to the cloud. In a Fog scenario when the computational elements are distributed and also heterogeneous, a policy is proposed in (9) to deploy applications over the Fog nodes in the form of "Droplets" which are smallest parts of the applications that can be placed in a distributed manner on resource constrained Fog nodes.

2. Dynamic node discovery in Mobile Fog (10): This second approach is based on a hierarchical orientation of Fog nodes and the run time scalability of distributed applications. Mobile Fog basically provides API that initiates dynamic node discovery process to send sensed data towards the vertically and horizontally placed nodes.

2.6 Distributed Data flow Approach

When the applications are distributed, and the application logic during execution can be represented a directed graph, Fog resources can be allocated according to the situation from time to time.

Latency Aware Fog Application Management

In cloud environment a latency – aware iterative algorithm is proposed (4) for co-deployment of cloud applications. The algorithm is repeated periodically as follows: first deploy applications in computational instances temporarily. Then, based on latency – sensitivity of applications, it iteratively identifies the appropriate instances. This is repeated periodically as the number of application users and their frequency of access can change in course of time. Further (5) proposed a latency aware application deployment for mobile cloud computing. A developed resource sharing architecture is proposed. A coordinator manages all incoming requests and resources in order to optimize server latency of various different applications

2.7 Resource Coordination Approaches

Resource discovery and Resource Scheduling is coordinated in an optimal manner by the coordinator. Various approaches and algorithms exist for this coordination. In Complex Event Processing (CEP), an algorithm for efficient application migration is prepared (6) based on plan based operator migration. Firstly, a time graph model is created to identify the possible migration targets and select one target instance based on the shortest path from source, and then runs the coordination algorithm to accommodate the migrating operation in the selected instances.

For communication latency aware application migration in a virtualized data centre, a Migration Manager identifies (7) communication patterns among virtualized instances at the time of migration. In addition, the Migration Manager is responsible for sorting the virtualized instances based on their current traffic, selecting an appropriate instance as migration target and checks its availability.

A scheduler solver is proposed (10) for scheduling of application in virtualized instances based on a concept of differential frequency multi slicing. Unlike other schedulers, solver divides a CPU slice into many micro slices and then it schedules in higher frequency according to micro slices. Thereby it increases CPU access probability of applications.

All the above are originally cloud based approaches, even though their concepts and logic can be utilized for Fog based application scheduling and management approaches.

3. FOG ANALYTICS

3.1 Introduction

While IOT contributed in a big way to Big Data, Fog Computing architecture becomes essential for IOT. If big data technologies depended entirely on the cloud, it is now very clear that it will be operationally infeasible, given the high volumes of data (coming out of IOT sensors and devices) and the high latency of the networks and internet bandwidth required for the huge data to reach the Cloud from the edge and coming back to the edge with decisions derived out of analytics, in the cloud.

3.2 Fog Computing

Fog computing is a new technology paradigm to reduce the complexity, scale and size of the data actually going up to the cloud. Pre-processing of raw data coming out of the sensors and IOT devices is essential and it is an efficient way to reduce the load of the big data on the cloud.

The Fog server, located very near to the edge devices, offers the possibility of pre-processing and even completing local analytics, to take fast decisions for the real time local edge requirements. Only the aggregate or summary data, small in size, needs to be sent to the cloud. This will lead to the benefits that accrue from Fog computing that include local, fast processing, storage for geo deductible and latency sensitive applications, drastically reduced communication overheads over the network and the internet, thereby having a substantially reduced volume and velocity of data that will be required to be sent to the Cloud.

Applications such as Augmented Reality, Interactive Gaming and Event Monitoring require data stream processing, a processing scenario in contrast with a ready data bank, assumed to exist in conventional big data application ecosystems.

3.3 Stream Data Processing

Stream data, arriving out of sensors and IOT devices is characterized by huge volumes of continuous data, fast changing, requiring fast real time response- data streams cannot be stored by virtue of their sheer size. Analytics on the entire data to be stored may not be feasible. Stream data examples include sensor data, IOT device data, RFID data, security monitoring, telecom call records, web logs, web pages clicks, credit card transaction flows, network monitoring, traffic data, stock exchange data all consider stream data. Data streams are characterized by their transient nature, with continuous queries, with sequential access, bounded main memory, in–memory data base management systems, and multi GB arrival rates having real time response requirement.

3.4 Stream Data Analytics and Fog Computing

Stream data mining and analytics and real time data mining provide for theoretical foundations of distributed stream data mining such as feature extraction and classification for Fog systems also. Recent open source software products such as ”Tensor flow” significantly facilitate the implementation of advanced data mining and machine learning algorithms such as deep neural networks in the Fog Servers and even in mobile edge devices (“Mobile Tensor flow“) Inspite of such implementations, the open unsolved challenges do exist – how to perform load balancing among multiple Fog Servers and edge devices without a affecting the performance? In addition, we have the well-known stream processing engines such as Apache Storm and Spark streaming that can be executed on Fog Servers. Therefore, we can say that we do not have a need for developing new tools for Fog analytics. Even then, the outstanding issues yet to be addressed are: APIs for Fog Streaming. While we do have Apache Hadoop ecosystem components’ such as Apache Mahout for machine learning and Spark GraphX for graph processing and that they can be used for Fog Streaming applications, the APIs for a few specific application or missing – differential equation solvers and control error estimators for Fog based real time network controlled applications.

3.5 Different Approaches of Fog Analytics

In the following sections we present a survey of the different approaches for Fog Analytics.

A. “Smart Data”

"Smart Data" is a package of encapsulated structural data generated by IOT devises and sensors, a set of metadata and an accompanying Virtual Machine (VM)-proposed for Fog Analytics.

B. Fog Engine

Fog Engine is an end to end solution that provides on- premise Data Analytics and communication capability to communicate with each other and also with the Cloud. Fog Engine is said to be customizable, agile and heterogeneous platform that is integrated to an IOT device. It enables data processing in the Cloud and in distributed grid of connected IOT devices located at the edge of the network. One Fog Engine can collaborate with other Fog Engines in the near vicinity, thereby creating a peer – to – peer network beneath the Cloud. It provides a facility for offloading the data and also for interacting with the cloud as a gateway. In-stream data is analyzed locally in a Fog Engine while data of multiple Fog Engines is collected and transmitted to the cloud for offline global data analytics, as may be required. Several scenarios can be identified for Fog Engine deployment, depending on multiple receivers, multiple or single analysers, multiple or single transmitters. The Fog Engine deployment can partially under take the burden of network back bone and data analytics at utilities side and reduce the dependence on the Cloud. While computations are done locally, only a fraction of the data that is cleaned and analyzed by the Fog Engine is transferred to the Cloud, thus drastically reducing the volume of data transferred over the network, resulting in substantially reduced network congestion and delay due to latency.

C. Other Products

Other Products in Fog Analytics include Microsoft Azure Stack and also Cardio log Analytics by Intlock which offers on- premise Data Analytics. Oracle delivers Oracle

Infrastructure as a Service (IaaS) on-premise with capacity on demand that enables customers deploy systems based on Oracle in their own data centres. IBM's Digital Analytics on-premise is the core Web Analytics software component of its Digital Analytics Accelerator solution.

D. Parstream

Parstream by CISCO enables immediate and continuous analysis of real time data as it is being loaded. Parstream has many attractive features as follows : highly scalable distributed hybrid data base architecture to analyze billions of records at the edge patented indexing and compression capabilities that process data in real time and also minimize performance degradation; makes use of standard CPUs and GPUs to execute queries; integrated with R language and other machine learning engines to support advanced Analytics- uses time series analysis for analyzing streaming data with massive amount of historical data – alerts and action are used and raised to monitor data streams, create user friendly procedures that generate alerts, send notifications and execute actions; derives models and hypotheses from huge amounts of data by applying statistical function and analytical models, using advance Analytics.

3.6 Comparison

All the above products have their own respective strengths and weaknesses. Although all of them offer on-premise data analytics services, they lack in providing a holistic approach based on the Fog Concept which is the intermediate layer between the Edge and the Cloud.

3.7 Cloud Solutions For The Edge Analytics

Solutions by Cloud Services Providers (CSPs) such as Amazon are also available – Amazon's AWA IOT offers implementing data collection through HTTP, Web sockets, MQTT and integrate with REST APIs with device gateway in Cloud. Amazon Quick Sight is available for Machine Learning purposes.

Microsoft offers Azure IOT Hub using HTTP, AMQP, MQTT and also custom protocols for data collection; offers REST APIs integration; offers stream analytics and Machine Learning, uses Azure IOT gateway (in-premise gateway).

IBM offers IBM Watson IOT using HTTP and MQTT for data collection, integration with REST and real time APIs. Data Analytics is offered through IBM's Blue mix Data Analytics platform.

Goggle offers Goggle IOT uses HTTP only for data collection, integrates with REST APIs and RPC: Analytics is offered through Cloud Data flow, Big Query Datalab and Dataproc and uses general gateway (on- premise) to Cloud.

Alibaba offers Alicloud IOT, uses HTTP, integrates with REST APIs, uses own Analytics solution, Max Compute and uses Cloud gateway to the Cloud.

In this section a survey of the approach, techniques and products for Fog Analytics are presented.

4. FOG SECURITY AND PRIVACY

4.1 Introduction

IOT devices form a Fog. Therefore the security and privacy issues in Fog are quite different from those of the Cloud. Data is better protected if stored in Fog servers than when it is located in the IOT devices. The Service Providers of Fog Services also will be different parties depending on the different deployment choices of Fog: Internet Service Providers or Wireless Carriers who have control of home gateway or cellular base stations may build Fog with their existing infrastructure. Cloud Service providers who want to expand the Cloud up to the Edge also may build the Fog infrastructure .End users who own a local private cloud and want to reduce the cost of ownership would like to turn the local private cloud into the Fog and lease the spare resources on the local private cloud. Accordingly, a trust model is required to be built by the Fog Service Providers.

4.2 Secure Communications in Fog Computing

IOT devices interact with fog nodes only when they need to offload a processing or storage request. Any other interactions would not be considered as a part of the fog environment as such communications would happen as part of the network. These fog nodes interact with each other when they need to effectively manage network resources or to manage the network itself. They may even operate in distributed manner to perform a specific task. To secure communications in a fog computing environment the following communications between these devices are to be secured:

1) Communication between constrained IOT devices and fog nodes and
2) Communication between fog nodes.

Usually, an IOT device can initiate communication with any of the fog nodes in the fog network requesting for a processing or storage requirement. In fact, the IOT device may not even be aware of the existence of the fog network, therefore messages sent by such a device cannot be secured by using symmetric cryptographic techniques. Alternatively, asymmetric key cryptography has its set of challenges that are unique to IOT environment. Maintaining the PKI (Public Key Infrastructure) that is required to facilitate secure communication is one of the major challenges. Other challenges include minimizing the message overhead keeping in mind the constrained environment in which the IOT devices operate. Communications among fog nodes requires end to end security as nodes involved in multi-hop path may not be trust worthy.

4.3 Authentication

The main security issue is authentication at various levels of fog nodes. Traditional PKI based authentication is not scalable. Near Field Coins (NFC) can be used effectively in Fog computing to simplify authentication procedures. Biometrics based authentication (such as the Aadhaar card in India) can also be effectively used in Fog computing ecosystem. Authentication becomes important at various levels of gateways or at the level of device itself. Each device such as a meter in Smart Grids or such as an i-pad in any Fog environment should have any authentication biometric based authentication or otherwise to prevent misuse, manipulation or impersonation. For example, Smart meters can encrypt the data and send to the Fog devices such as Home-Area- Network (HAN) where the data can be decrypted, the results be aggregated and then pass them forward to the cloud, for example.

Traditional authentication mechanisms such as PKI (Public Key Infrastructure) are not suitable due to resource constraints of the IOT devices. Alternative protocols such as 3GPP, GBA, OMA, M2M are proposed to be deployed.

4.4 Privacy Issues

Privacy issues pertaining to the devices-which device was used, when, for what purpose, etc are required to be analyzed. Encryption can be used to provide encrypted result which cannot be decrypted by the individual devices.

4.5 User Behavior Profiling

Insider data theft can be prevented by user behaviour profiling in Fog. To detect abnormal user behaviour, the current user behaviour can be compared with standard user behaviour to find out deviations, if any.

4.6 Dynamic fog nodes and EUs

Similar to mobility issue in EUs (End Users), the fog nodes also frequently join and leave the fog layer. It is required to ensure the uninterrupted service to the registered end users when a new fog node joins (or leaves) the fog layer. The EUs must be able to authenticate themselves to the newly formed fog layer mutually. From EUs perspective, the complexity of registration and re-authentication phase is a huge overhead.

4.7 Malicious Attacks

Fog computing environment can be subjected to several malicious attacks and without proper security measures in place may severely undermine the capabilities of the network. One such malicious attack that can be launched is a Denial-Of-Service (DOS) attack. Since majority of the devices connected to the networks are not mutually authenticated, launching of DOS attack becomes straight forward. The attack may be launched when devices that are connected to IOT network request for infinite processing/storage services. That is, a compromised or malfunctioning node can make repeated processing/storage requests to a fog node thereby stalling requests made by legitimate devices. The intensity of such an attach rises manifold when a set of nodes simultaneously launch this attach. Another way to launch this attack is to spoof addresses of multiple devices and send fake processing / storage requests. Existing defense strategies of other types of networks are not suited for fog computing environment mainly due to the openness of the network. The first major challenge is the size of the network. Potentially, hundreds and thousands of nodes forming an IOT network avail the services of fog/cloud to overcome computation and storage limitations and also enhance performance. Since all these devices cannot be authenticated by fog nodes, they may rely on trusted third party like a certification authority that issues some form of credentials to ensure device authentication. But, the existence of such credentials only allows the processing fog node to verify whether the request has been generated by a legitimate node.

Since a compromised node is a legitimate part of the network, all such requests would be entertained. On the other hand, restricting connectivity to the network or filtering the requests made by IOT devices nullifies the motivation of existence of fog nodes. Spoofing of addresses is also relatively easier as the address space is relatively large and lack of boundaries makes it even more difficult.

4.8 Malicious Insider in the Cloud

One of the severe attacks to the Cloud computing is the data theft attack by a malicious insider to the Cloud provider. Basically, the end users have to trust the Cloud Service Provider. Thus, lack of Cloud Service Provider's authentication results in data theft. Many incidents such as Twitter's personal and corporate data hacking and ex-U.S. President Barack Obama's account hacking reveal that the end user's password can be stolen effortlessly by a malicious insider. In cloud computing the malicious attacks are prevented by using user behaviour profiling. In Fog, the design issues are:

Where to place decoy in fog network?

How to design on-demand decoy information to further reduce the amount of stolen data?

Side Channel Attack: An attacker could attempt to compromise the cloud through placing a malicious virtual machine in close proximity to target the cloud server and then exploiting a side channel attack.

4.9 Man in the Middle Attack

The Fog is vulnerable and the Man in the Middle is an example of this vulnerability. In this type attack an attacker places himself between two parties in a communication network. In this situation the gateway serving as the Fog device may be compromised and replaced with fake or malicious access paths which provide deceptive SSIDs as public, legitimate ones. Any time attacker can place himself in the communication path, intercept and modify the messages passing through the path. Thereby the attacker can take control of the gateways and thus the private communication will be hijacked. Man in the Middle attack in Fog computing can be very stealthy. It is very difficult to protect the Fog devices from it. A user may be able to obtain information from the attack but have to decrypt the information before it can be read.

4.10 Secured Multi-Tenancy

Fog computing should enable highly constrained access control on both data and network, along with fair resource allocation mechanisms to protect confidentiality and integrity without a multi-user environment.

When a large number of end-users start to share Fog applications and resources, the performance, scalability, data security, user identity management, monitoring and the potential arising from insiders threats becomes difficult to manage in a Fog network. Fog platforms should implement multi-factor authentication mechanisms based on either the role or identity of end users, logically segregate data and resources and aggressively analyze the activities of both administrator and tenants. Another system called Secure and Resilient Networking (SeReNe) service can provide a Fog platform with programmable environment to adjust it's topology, bandwidth allocation, and traffic policies.

4.11 Backup and Recovery

Depending upon the kind of application, Fog platforms should have data backup and recovery modules. Such system should mirror copies of data on-site, off-site or both on a regular basis. It will benefit both customers and company to keep the operations running from using previous backups, minimizing service disruptions.

The Fog platform has a high frequency of data throughput and relatively low amount of stored data, but this does depend on the requirements and application. The challenge is that data backup and recovery is a costly process and requires acute focus on data selecting, mapping, testing and determining accessibility roles in case of recovery process. In case of natural disaster, system failure or cyber-attack, Fog platforms can lose all data and hence there is a need for primary and secondary backups. It is important to not duplicate the data before backup. It will decrease costs and notably reduce the consumption of resources during backup process and recovery. There are also many methods available to improve the process in terms of consistency, co-ordination and performance, such as Fibre Channel, High Security Distribution and Rake Technology (HS-DRT), Parity Cloud Service technique (PCS), Efficient Routing Grounded on Taxonomy (ERGOT), Cold and Hot Backup Service Replacement Strategy (CBSRS) and Shared backup router resources (SBBR). For mobile and wireless Fog platforms, the situation might get challenging as the system would require portable and on-site backup storage or will need a significant amount of network bandwidth to transmit data to the off-site location.

5. RESEARCH DIRECTIONS

The following are the emerging directions of research in IOT and Fog Computer area:

1. Harnessing Temporal Dimension of IOT Data For Customer Relationship Management (CRM)

   The temporal perspective of IOT data can be harnessed to help settle continuous choices, enhancing quality of service and quality of experience by providing an amazing experience of client encounter – For example, a buyer arranged association can join the accessible customer information with every day shopping lot occupation information, to offer motivating ideas to their customers or deal with their stock in a proactive way, on a day to day basis or on occasional basis. This pattern changes into a proactive one to influence relationships to determine bits of knowledge, discover new relationships managing well the existing or old relationships with customers (CRM).

   Analytics solutions such as regression techniques from Machine Learning and Deep Learning can harness the temporal dimension of IOT data.

2. Adding Semantics to IOT data

   The value of information can be enhanced by adding Metadata to their specific circumstances and implications. Such metadata will be important to IOT data by helping clients process and use heterogeneous IOT data at the edge or device level or at Fog level.

   Vocabularies of metadata are to be deployed to create ontologies. Ontologies will be useful to combined or for merging the IOT information from different spaces. Efforts will be required for making specific ontologies for connecting and sharing information across different areas of applications and organizations. Ontologies can be based on RDF, RDMS and OWL or OKM by integrating URIs.

3. Towards a Semantic web of IOT

   Once ontologies are well defend cutting cross all different applications and domains, a semantic web of IOT can be targeted as an objective where the multimedia data coming from IOT devices and sensors will be integrated into the semantic web.

4. Diversity, Interoperability and Standardization im IOT

   The IOT world view has heterogeneous traditions, standards, stages, and initiatives used as CoAP, MQTT, XMPP, STOMP, HTTP 7 AMQP. Integration between them is essential. Vendor lock-in is along with platforms such as IBM Watson, Microsoft Azure, GE Predix, Cisco Jasper and QTC ThingWork. Open source IOT arrangement exercises such as "things board. Io", 'Kaa', Device Hive are good attempts in these directions.

5. Data Management Issues in IOT

   New mechanisms such as Data Lakes have come up to deal with huge information coming out of IOT. An Information Lake stores organized unstructured information with no pre imagined idea of how these information items may be used subsequently. In a data lake issues as data quality, meta data and reliability become essential.

6. Data Provenance

   Information provenance is connected to the genuineness and respectability of the information, also to the traceability to decide the owners / proprietors and data modifiers of information at each progression. Given that huge information gives profound experiences and investigations that may prompt some type of autonomous activation, in reality, we should guarantee that the information items utilized for making such important decisions are originating from real and dependable sources. Information provenance issues in IOT are to be addressed, in addition to the issues of information administration.

7. Data Governance and Regulation.

   The IOT data originally from a variety of sources such as private homes, roads, farms, factories. It needs to be governed and administrated. Information Administration and controlling its utilization needs to be executed effectively. The owners of IOT gadgets should be given the rights to indicate their preferences, intentions and instructions to permit the utilization of the data originating from their respective IOT gadgets. Future research should bring out the possible creation of arrangements and structures to recognize the stakes and concerns of the proprietors of IOT information, the consumers of IOT information and also other actors who act between these two.

8. Context aware Resource Provisioning and Context aware Service Provisioning.

   Context awareness can lead to efficient resource and service provisioning in Fog Computing – Context can be

   - Environmental context: location, time (peak or off)
   - Application context: Latency sensitivity, & Application Architecture etc.
   - User context: Mobility, Social Interaction Activity etc.
   - Device context: available resources battery life time, etc.
   - Network context: Bandwidth, network traffic etc.

9. Sustainable and reliable Fog Computing

   The sustainable architecture in Fog Computing is subject to many issues like assurance of QoS (Quality of Server) QoE (Quality of Experience), Service reusability, energy efficient resource management etc. As far as Reliability is concerned, it is dependent on consistency of Fog nodes, availability of high performance services, secured interactions, and fault tolerance.

   Research in this direction is having great potential towards achieving sustainable and higher performance level in Fog Computing.

10. Interoperability among Fog Nodes

    Fog nodes function both as network nodes and computational nodes. In addition to performing traditional networking activities like packet forwarding, packet switching, routing, forwarding, etc. Fog nodes perform computational tasks also, depending on the context of the role they play. They also need to have Interoperability. Therefore an interoperability architecture of Fog nodes that can be self-customized accordingly to the requirements is essential in literature many unique architectures of Fog nodes can be found. Till now no real interoperable architects of Fob nodes has emerged.

11. Distributed Application Deployment

    Fog nodes are distributed and they may have heterogeneous resources. While in literature there have been proposed several programming platforms for distributed application development and deployment, issues such as latency based application management optimization, data flow management, assurance of QoS and QoE, edge centric affinity etc are open problems.

12. Power Management within Fog

    As the number of active Fog nodes increases (based on demand) the power management within Fog network becomes critical.

    Till now only power saving techniques for cloud data centers have been investigated. Optimization of power within Fog network is an open problem. For this purpose migrating tasks from one node to another way become necessary. Research is required to come out with optimization strategies for power management within Fog networks.

13. Multi Tenancy support in Fog

    The Fog resources can be virtualized and allocated to multiple users. This area of multi-tenant support in Fog resources and scheduling the computational tasks according to QoS requirements is open for research.

14. Programming Language and Standards for Fog

    Fundamentally the Fog has been proposed to extend cloud services such as IaaS, PaaS and SaaS to the edge. Since the Fog differs substantially from cloud, the programming language and standard for Fog will be quite different from those of the cloud. It is highly essential to identify new programming languages and standards for the Fog environment.

15. Simulation in Fog

    "Fog Sim" has been developed (at University of Melbourne) for simulating Fog eco system. However the development of new efficient simulations for Fog computing can be a future possibility.

16. Mobile Fog: Research Opportunities

    1. Proactive Vs Reactive Service Migration

       When IOT nodes physically move away from their associated Fog service, the benefits of Fog computing need to be continued to be preserved. For this purpose the Fog service has to be kept topologically close to the mobile or moving devices or IOT nodes. That means the horizontal or vertical handover (or migration from one station to another) of the mobile nodes is the actual event capable of causing a substantial change in topological distance. Therefore migrating decisions (as to where and when) should be made proactively or reactively, considering this event as corner stone, depending on the random mobility of the users. Research is required to develop algorithms in this scenario.

    2. Exploiting context information to target service migration

       Which parameters determine migration? They are context dependent, varying from case to case and also depend upon actual objectives such as QoS, bandwidth, etc. The parameters such as network state, 'fog nodes' state, amount of data exchanged by the application and migration costs in terms of bandwidth consumption and time taken. It will be of benefit to take the context into consideration as a parameter. (eg. In a hospital scenario patient's location within the area of hospital premises can be considered by Fog nodes belonging to the hospital among the eligible target nodes).

    3. Fog federation for enabling mobile roaming

       When a mobile user moves from one location to another where the original Fog node does not exist, the Fog Federation approach way be able to help better. A Fog node that belongs to a federated domain will offer a better performance alternative. Federation is a trick that can provide a user with a considerably closer Fog node that belongs to the same federated domain (to which the previous node also belonged) challenges involved in forming and managing Fog federation have not yet been addressed in research. Some such challenges are (i) management of SLAs among Fog nodes, (ii) Architecture for Fog federation in distributed or peer to peer and (iii) which technologies to harness.

    4. Virtualization and Migration Techniques

       The performance of migration will depend upon carefully choosing virtualization and migration techniques. For example, to host a Fog Service, in order to expand the set of nodes qualified for this purpose, it is essential to find a virtualization technology which is suitable to as many physical nodes as possible. Also it is equally important to identify a migration technique to be able to minimize the overall duration of migration and of subsequent down time and also minimize bandwidth consumption. Scope of research exists for all these issues.

5. Integration with mobile networks, towards 5G

   Mobile Edge Computing (MEC) is a standard Fog computing system depend by European Telecom Standard Institute (ETSI). It is considered a key enabler for IOT services and of upcoming 5G network. Interacting research direction is to prepare mobility support solution as part of ETSI, MEC system thereby enabling mobility and session management interfacing. For example Mobility Management Entity (MME) component of Long Term Evolution (LTE) core network provides some useful functionalities such as (1) retention of users' location information, (2) selection of appropriate gateway during initial registration process (NSN) management of handover between LTE and 2G/3G networks. (3) management of roaming from other LTE or legacy networks (4) Session Management. All the above are to be well integrated in an optimization strategy by appropriate research.

CONCLUSION

In this survey we have comprehensively covered all the aspects of Fog Computing as Fog Application Management, Fog Analytics, Fog Security and Privacy and also the research directions.

REFERENCES

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems 29, 7 (2013), 1645-1660.

[2] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya, 2016. iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments. ArXiv preprint arXiv: 1606, 02007, Technical Report CLOUDS-TR-2016-2, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne (2016).

[3] Nam Ky Giang, Michael Blackstock, Rodger Lea, and Victor CM Leung, 2015. Developing iot applications in the fog: a distributed dataflow approach. In Internet of Things (IOT), 2015 5th International Conference on the, IEEE, 155-162.

[4] Y.Kang, Z. Zheng, and M.R.Lyu, 2012. A Latency-Aware Co-deployment Mechanism for Cloud Based Services. In 2012 IEEE Fifth International Conference on Cloud Computing. 630-637. DOI: http://dx.doi.org/10.1109/CLOUD.2012.90

[5] Takayuki Nishio, Ryoichi Shinkuma, Tatsuro Takahashi, and Narayan B.Mandayam. 2013. Service-oriented Heterogeneous Resource Sharing for Optimizing Service Latency in Mobile Cloud. IN Proceedings of the First International Workshop on Mobile Cloud Computing &#38; Networking (MobileCloud'13). ACM, New York, NY, USA, 19-26, DOI:http://dx.doi.org/10.1145/2492348.2492354.

[6] Beate Ottenwalder, Boris Koldehofe, Kurt Rothermel, and Umakishore Ramachandran.2013. MigCEP:Operator Migration for Mobility Driven Distributed Complex Event Processing. IN Proceedings of the 7th ACM International Conference on Distributed Event-based Systems (DEBS'13). ACM, New York, NY, USA, 183-194, DOI:http://dx.doi.org/10.1145/2488222.2488265.

[7] Takouna, R.Rojas-Cessa, K.Sachs, and C.Meinel. 2013, Communication-Aware and Energy-Efficient Scheduling for Parallel Applications in Virtualized Data Centers. In 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing. 251-255. DOI:http://dx.doi.org/10.1109/UCC.2013.50

[8] Md.Redowan Mahmud, Mahbuba Afrin, Md.Abdur Razzaque, Mohammad Mehedi Hassan, Abdulhameed Alelaiwi, and Majed Alrubaian, 2016, Maximizing quality of experience through context-aware mobile application scheduling in cloudlet infrastructure. Software: Practice and Experience 46,11(2016), 1525-1545, DOI:Http://dx.doi.org/10.1002/spe.2392 spe.2392.

[9] Luis M. Vaquero and Luis Rodero-Merino, 2014. Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing SIGCOMM Comput. Commun. Rev.44,5 (Oct.2014), 27-32, DOI:http://dx.doi.org/10.1145/2677046.2677052

[10] Kirak Hong, David Lillethun, Umakishore Ramchandran, Beate Ottenwalder, and Boris Koldehofe, 2013. Mobile fog: A programming model for large-scale applications on the internet of things. In Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing. ACM, 15-20.

REFERENCES (Web Links)

1 https://www.researchgate.net/publication/319442626_Security_and_Privacy_in_Fog_Computing_Challenges

2 http://www.engpaper.com/fog-computing-2016.htm

3 http://ijrise.org/asset/archive/CSE_UG510.pdf.

4 www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing

5 https://www.gsma.com/iot/gsma-iot-security-guidelines-complete-document-set/