♦ Recepción/ 27 junio 2019
♦ Aceptación/ 25 agosto 2019

Edge Computing: Review and Future Directions

Computación De Borde: Revisión Y Direcciones Futuras

Amjad Hameed Shehab[a], Sufyan T. Faraj Al-Janabi[b]

[a]Department of Computer Science, College of Computer Science and IT, University of Anbar Ramadi 55, Anbar, Iraq (e-mail: rkos.vaio@gmail.com)
[b]Department of Information Systems, College of Computer Science and IT, University of Anbar
Ramadi 55, Anbar, Iraq (e-mail: sufyan.aljanabi@uoanbar.edu.iq)

ABSTRACT/ By leveraging cloud computing architecture and services to do centralized computation, especially when Internet of Things (IoT) scenarios want to react from insights resulted from that computation back to end devices, then we run into actual limitations of bandwidth congestion and the resulted high latency. Edge Computing come to existence with different implementations to gradually remove these barriers on limitations. Current networks relate to edge computing success to make an advance in their services to end users such as the shift from current 4G network infrastructure to 5G enhanced one. This paper aims twofold: Firstly, a review of the concepts and technology of edge computing is given. This includes cloud computing, the emerging edge computing and its implementations, comparison between existing terminologies, and an overview of wearable devices scenarios. Secondly, the paper presents an investigation of two promising technologies that have been adopted to implement the edge computing setup in our project. These are Azure cloud service and Raspberry Pi edge devices. Keywords: Cloud computing; edge computing; fog computing; IoT; Azure services. **RESUMEN** **Al aprovechar la arquitectura y los servicios de computación en la nube para hacer un cómputo centralizado, especialmente cuando los escenarios de Internet de las cosas (IoT) quieren reaccionar a partir de los conocimientos resultantes de ese cómputo de regreso a los dispositivos finales, entonces nos topamos con limitaciones reales de congestión de ancho de banda y la alta latencia resultante. Edge Computing llega a existir con diferentes implementaciones para eliminar gradualmente estas barreras a las limitaciones. Las redes actuales se relacionan con el éxito de la informática de punta para hacer un avance en sus servicios a los usuarios finales, como el cambio de la infraestructura de red 4G actual a una mejorada 5G. Este documento tiene dos objetivos: en primer lugar, se ofrece una revisión de los conceptos y la tecnología de la informática de punta. Esto incluye la computación en la nube, la computación de borde emergente y sus implementaciones, la comparación entre las terminologías existentes y una descripción general de los escenarios de dispositivos portátiles. En segundo lugar, el documento presenta una investigación de dos tecnologías prometedoras que se han adoptado para implementar la configuración informática de borde en nuestro proyecto. Estos son el servicio en la nube de Azure y los dispositivos de borde Raspberry Pi. Palabras clave: Computación en la nube; computación de borde; computación de niebla; IoT; Servicios de Azure.**

Introduction

Two main advancements in networking technologies have made edge computing possible: Cloud computing and IoT. Cloud computing up grown in mid-2000s [1], which made the processing and storage far from user devices possible by leveraging high bandwidth and reasonably low latency connection to infrastructure. This had resulted in high capabilities of processing power and storage possibilities. The idea behind cloud computing is to separate the application, operating system, and hardware from each other. That means everything is virtual. When one underlying hardware component or cloud infrastructure fails, it does not matter because the infrastructure is separated from platform and that last is separated from software. Thus, the basic development models of cloud computing are the following [2]:

- Software as a Service (SaaS); google docs, Gmail, office 365 features are all software applications resides in cloud presented to us as a service. In other words, if we want just a software to use it then we choose SaaS development model.
- Platform as a Service (PaaS); almost adapted to programmers to write and run their codes in the cloud. Microsoft Azure is a platform as a service model that make use of cloud services along with codes that we wrote to develop web-based applications. In other words, if we have software but need a platform to deploy and run it then we choose PaaS development model.
- Infrastructure as a Service (IaaS); instead of having telephone systems in our buildings or user-configured security system on our own, we simply buy this hardware as a service from providers and cut cost and time of provisioning and maintenance. In other words, if we have a software and platform and we need a hardware to run those then we use IaaS development model.

The generation of edge computing as a disruptive technology will change our look to near future. Edge computing has the potential to offload essential computing and storage resources to edge nodes that can represent a small data center; a smartphone or a PC with proximity of one wireless hop away from real devices. Many people believe that edge computing is disruptive technology not an incremental because of these fundamental reasons [1], [3]:

- Highly responsive cloud services so low latency will be assured.
- Edge analytics in Internet of Things (IoT) so bandwidth mitigation on network is forced.
- Exposure security system in IoT in which user will become the responsible for configuring security system on its part of network so privacy is checked.
- Mask disruption of cloud services in which no more intermittent connection caused by bandwidth congestion to be a problem so availability will be presented.

Furthermore, edge computing provides cloud services in powerful way for IoT devices. It also provides security, privacy, and user-configurable security system along with the ability to avoid cloud outages caused by bandwidth congestions and hence intermittent connection. Edge computing is a promising technology because it interacts with and used effectively in assembling, storing, and analyzing data from real devices to making IoT devices well user-configurable in terms of security, privacy, etc. This can effectively move the barriers that prevent, for example, vehicular system from leveraging cloud services to do more intelligence with low latency. When such technology is added to current architecture and infrastructure of network, a significant mitigation of bandwidth congestion is guaranteed and hence low-latency-dependent IoT devices will become more promising and realistic. A quality of service (QoS) enhancement, reduced transmission cost, and semi-storage on edge nodes are what people are waiting from such technology to implement efficiently.

The remaining of this paper is organized as follows: Section II defines edge computing briefly along with its main challenges. Section III represents a comparison of the basic edge computing implementations. Next, Section IV discusses the impact of wearable devices on IoT and relation to edge computing. Some interesting related works in the literature are surveyed in Section V. Then, we report on our ongoing edge computing platform development in Section VI, where the two main technologies adopted in this project which are the Microsoft Azure Platform and Raspberry Pi are more thoroughly considered. Finally, Section VII concludes the paper.

Overview of Edge Computing and Its Main Challenges

In this section, some basic concepts of edge computing are reviewed. Also, the most important challenges of edge computing are introduced.

Edge Computing Basic Concepts

Edge computing has emerged with the proliferation of Content Delivery Networks (CDNs) that aim to speed up web performance by caching web contents on edge nodes near user [1]. Edge computing can show more features than CDN concept do by running stream of code just the same that found in cloud computing. Edge computing can leverage from applications that are sensitive to latency limitations.

Figure 1 shows an implementation of Cloudlets (will be more emphasized shortly) in edge computing when videos from mobile device do not need to travel far from its source to be processed but only to near cloudlet node. The results only sent to cloud if necessary. This significantly reduces the ingress bandwidth into the cloud. By putting all technologies together, as Figure 2 shows, the N-tier architecture for edge computing is formed.

Edge Computing Challenges

The most important challenges of edge computing are described in the following subsections.

Programmability

When a program service or application is needed in the cloud, we issue code written in specific language targeting a specific platform and deploy it to cloud. At this end we no longer know how our service or application runs inside the cloud, i.e. in which cluster hosted since virtualization is originally enabled in cloud. In the edge computing scenarios, nodes characteristics in their runtime are heterogeneous. Thus, the service or application needs further dependencies; libraries offloaded with it to run at nodes with said heterogeneity of the environment. Hence, it is prudent to propose the concept of computing stream that shows multiple computational handlers such as the functions conducted on the data along its traveling to a destination. These handlers may process data on demand which means that application decides which to operate fully or partially according to some issues such as device constraints. The applied computing stream will significantly reduce the latency by processing data close enough to the source [5].

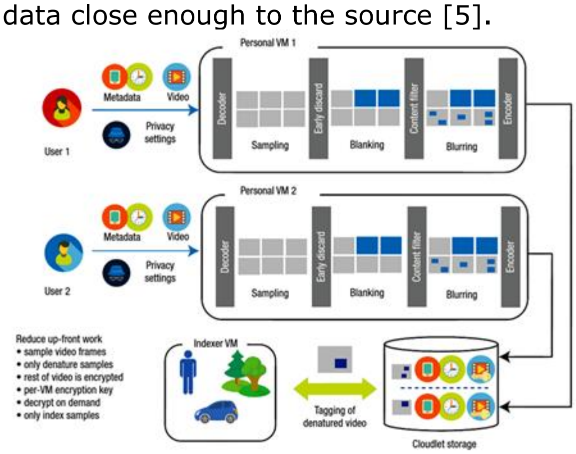

Figure 1.A sample edge computing implementation (GigaSight framework) based on cloudlet to perform computer vision analytics [1]

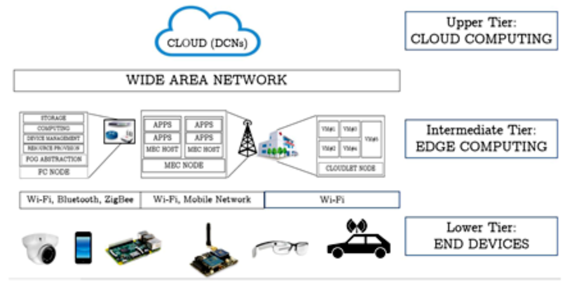

Figure 2.N-tier Architecture for Edge Computing Paradigm [4]

Naming

With all these large number of edge-served devices in edge computing, an efficient yet simple naming mechanism is mandatory in these circumstances. In each of these edge-served devices there are a lot of applications that each has different structure due to its role in device. Such requirements of efficiency and reliability in naming standards are inherited and worked well in some traditional techniques such as DNS and uniform resource identifier. At some circumstances where the mobility of things in edge computing is highly addressed, the previous mechanisms considered to be in malfunction. The researchers must understand and take into accounts the privacy protection of data, how to deal with heterogeneity of devices through simplifying network topologies and protocols, and address the mobility issue afterward. D. Raychaudhuri et al. [6] introduced MobilityFirst as naming mechanism that could be applied to edge computing by separating name and network address from each other to support mobility nature of devices. Another approach in naming in small edge environment such as smart home is edgeOS [5]. Attaching network address to each thing in such environment can be a responsible solution. For example, an oven in kitchen with temperature sensor is translated into "kitchen. oven2.temperature3." This helps users to identify any error or failure in that specific device directly instead of having to search for the error through error code. Figure 3 shows the edgeOS naming mechanism.

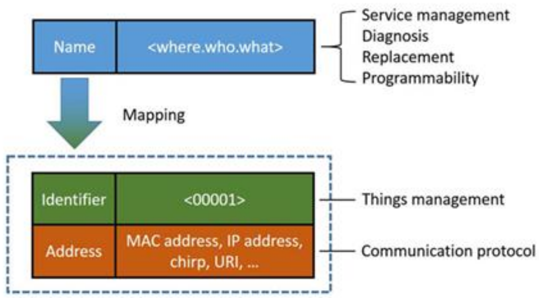

Figure 3.Naming mechanism in edgeOS [5]

Data Abstraction

In IoT, almost all things around could generate data and send it to gateway for processing. For example, consider a smart home scenario and edgeOS on top of it. Data abstraction is the challenge of processing data, literally kind of understanding the raw data, close to source so that to assure security and privacy before can be transmitted to higher level or stored. Assuming a thermostat that sends temperature every minute and a security camera which sends video of the place around; in both cases, the data will be stored for some period and some data arriving will get in place and old data removed, i.e. resource-constrained devices. Thus, one can imagine the data abstraction challenge regarding processing data for noise removal, event detection, privacy concerns, etc. [5].

First, data coming from devices usually are in various formats, as shown in Figure 4. Applications used to consume data should only consume data interested in. This requires making a table for each entry representing information such as id and name of device, time, and data from which applications can query. However, in this case, the usability of data may be affected. Second, the degree of data abstraction at various cases is difficult to find. Sometimes, conducting filtration on data will cause losing the meaning of data at higher levels as data continues to travel along. Alternatively, more quantities of data needed to be stored in devices will be another challenge in space acquisition. Lastly, the things operating in hostile environments at general may generate data that considered unreliable.

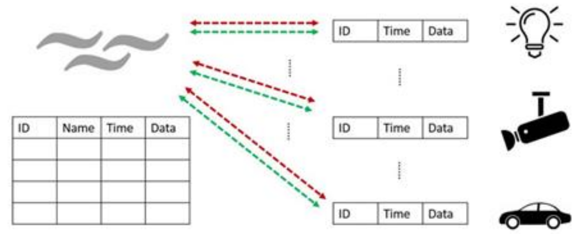

Figure 4.Data abstraction issue for edge computing [5]

Service Management

To consider an edge computing system is reliable, it needs to fulfill the following fundamentals [7]:

- Differentiation—when edge computing is evolving, there are many services ready to be deployed to edge device to service multiple user demands. Priority in processing information by these multiple services should be considered for processing from high priorities such as alert and health care systems to low priorities services such as entertainment.
- Extensibility—when we need to replace and old device/node or adding new one, the designer of that layer must consider the high mobility nature of things as well as the reasonable easiness presented to users during production.
- Isolation—this challenge stands for separating the edge device runtime from malfunctioning application running on top of it. For example, if the application that can turn the lights on and off seems to be not responding, the runtime should not be affected by this behavior. Instead, another application that also has access control to lights resources should continue to work afterward.

Privacy and Security

One of the most challenges that keep edge computing open to potentials vulnerabilities is security and privacy of data collected and being collected [8]. Take smart home as an example. An intruder may hack the edge system and easily uncover that the home hosts an elderly person through learning from some data that are being used by healthcare application inside edge device runtime. Thus, these challenges in privacy and security remain open for research. One of these challenges is awareness of the community to privacy and security concerns.

Most of household routers are still today configured with default password. If edge computing applied with this misunderstanding at surface, some of wifi-enabled toys that still configured with default passwords will be a serious problem. Another issue of privacy is when things are being able to kind of harvesting very private data, i.e. readings from user body. The user have to decide when to store and when to forward data after removing some privacy concerns from it to service providers site for long term storage. The third and most important issue to consider is to remember that almost edge devices are resource limited nodes. Some traditional, resource hungry methods for protecting user data are not applicable to the edge devices due to the previous limitation issue.

Application Distribution

As edge computing evolving, more computing power are estimated from such infrastructure. In most cases, applications distributed to the node as multiple components according to computing a resource, energy efficiency, and response delay of the said node [8].

Optimization Metrics

Allocations strategies are a must for deciding how to handle the workload at a different layer in edge computing implementations. One needs to consider several optimization metrics in terms of latency, bandwidth, energy, and cost [8].

Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing

There are optical networks that connect among cloud data centers to appear as a Data Center Networks (DCNs). Edge computing leverages from rich computation and storage devices capabilities in IoT. Hence, these edge node devices serve as an intermediate layer between end devices and cloud. In general, there are three types of edge layer implementations: Fog computing, Mobile edge computing (MEC), and Cloudlet computing. Implementation of edge layer between end terminal devices and cloud depends on many aspects such as devices that act as edge nodes, communication protocols and networks used by edge layer and services offered by edge layer [4].

In terms of functionality, Fog computing leverage from devices like wireless router and access point comprising edge layer, while MEC is leveraging devices with computing and storage capabilities and put them in the base station of traditional cellular network causing cloud computing services goes inside Radio Access Network (RAN). Cloudlets are small data centers with lower scale capabilities being deployed on user side utilizing computation offloaded from end devices. It is possible to make a comparison based of these edge computing implementations to address the gap from various existing literature as follows [3]:

- Fog computing: It uses Fog Computing Nodes (FCNs) as a computing infrastructure. FCNs are composed from many heterogeneous components such set top boxes, wireless routers, IoT gateways, and access points. The heterogeneity nature of nodes is not visual to terminal devices through the existence of Fog Abstraction Layer. Fog Abstraction Layer is a uniform and programmable interface for seamless resource management and control. The layer provides generic API's for monitoring, provisioning, and controlling physical resources such as CPU, memory, network, and energy.
- MEC: By using the RAN, edge computing in MEC is conducted on base station within RAN to decrease latency and being situation awareness.
- Cloudlet computing: A cloudlet expresses a small data center in a box with resources available to use by nearby devices.

Furthermore, each edge computing implementation needs to effectively handle various requests from applications such as requests from wearable devices. Thus, offloading operation from end devices in different edge computing implementations can be conducted as follows [4]:

- Fog computing: in fog computing paradigm, there are FCNs that communicate with Fog Orchestrator through function exposed by Fog Abstraction Layer. Typically, a request from device has an obligation statement such as certain QoS satisfaction. Fog orchestration check services introduced by nodes and fair trade-off are conducted between nodes to choose appropriate one accordingly.
- MEC: In MEC architecture, there is a number of hosts each running a number of applications inside it. Receiving of data is done through MEC servers that are located cooperatively with base station at the Mobile Edge Orchestrator. This

orchestrator works through running application and available resources of each to direct data to suitable node for processing, else it will direct data to cloud for processing.

- Cloudlet: The Cloudlet Agent oversees the cloudlets and the underlying nodes. The node agent and execution environment present capability of interaction between cloudlet agent and underlying components. The cloudlet agent then takes the responsibility of making good decisions to underlying nodes by means of policy violations directed by components hierarchically.

Table 1 represents a comparison among different edge computing implementations in terms of various edge computing elements including: Types of devices used, location of these devices in architecture, software architecture, context awareness, proximity from end devices, access mechanism, and internode communication.

Table 1.
Comparison of Edge Computing Implementations [4]

EC Elements	Fog Computing	Mobile-Edge Computing	Cloudlet Computing
Node Device	Routers, Switches, Access Points, Gateways	Servers running in base stations	Data Center in a box
Node location	Varying between End Devices and Cloud	Radio Network Controller/Macro Base Station	Local/Outdoor installation
Software Architecture	Fog Abstraction Layer based	Mobile Orchestrator based	Cloudlet Agent based
Context awareness	Medium	High	Low
Proximity	One or Multiple Hops	One Hop	One Hop
Access Mechanisms	Bluetooth, Wi-Fi, Mobile Networks	Mobile Networks	Wi-Fi
Internode Communication	Supported	Partial	Partial

Wearable devices and Edge Computing

The impact of wearable devices has grown in recent years and still growing. Smart watches and wearable cognitive assistance are game changer devices in our near future. Although these devices are tiny, resource-constrained in terms of computing power, short-term connectivity, and battery life, they introduce more comfortable life to us when connecting them to local-processing servers such as Cloudlets. Furthermore, they will introduce more useful assistance, functionality, and longer battery life as we offload computing-intensive processing to edge of the network. However, one should be careful as there may be circumstance where offloading processing to edge devices makes things more complicated as communication is much more expensive than computation.

For example, Google Glass is a smart cognitive assistance device. One of use cases is used to help a person to construct simple Lego parts into pre-defined structure by sending frames to Virtual Machines (VMs) residing in Cloudlet. Each VM does a specific task on frame such as face recognition and object recognition, then sends back helpful guidance reaching a person by glass microphone and displayed if necessary, on Glass visual layer [9]. Such applications are not much latency-intensive related.

Another task that can be done on the same device is directing a player in a ping-pong game to respond to her/his opponent coming ball. That kind of applications is much latency-sensitive, where video stream is offloaded to local server for processing and hints go back to Glass displayed on its Visual Layer. Also, voice instructions guidance can be introduced through device Mic. Figure 5 shows the backend processing on cloudlet for the scenarios stated above.

The wearable devices and hence cognitive assistance first impression was reported in 2004 when a thought piece imagined a world with real-time cognitive assistance for a mobile user [10]. In this prospect, many similar scenarios might be considered. One of these is to exploit features of transparent computing to build scalable IoT platforms and then build scalable lightweight wearable device leveraging the proposed architecture. Here, smartphones are used as wearable edge devices and a high-performance PC as cloud server [11], [12].

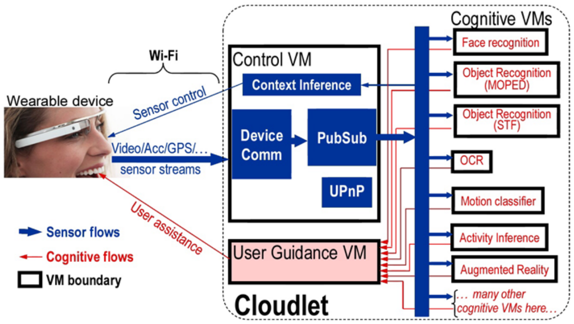

Figure 5.Back-end processing on Cloudlet [9]

Related Work

In this section, a review of some interesting related works in the literature is presented. These works are divided into two categories: A general category on edge computing and a more specific category including works on cloud offloading.

Edge Computing

Mahadev Satyanarayanan in [1] made an excellent historical overview for origin of cloud computing, its driving forces, and a comparison between CDNs, which used to cache web intensive content like videos to nodes near users, and emerging edge computing that extends that by leveraging of cloud computing infrastructure. The author also illustrated why the proximity of Cloudlets to end devices will help out by proving that through several distinct ways. Thus, he concluded that Cloudlets can increase the privacy of data. That is because they free users from over centralization of IoT by the ability of deleting sensitive data before sending along with the ability of cloudlets to masking cloud services in outages situations.

In order to realize the different implementations of edge layer between end devices and cloud, Koustabh Dolui and S. K. Datta [4] conducted a comparison between these edge computing classifications based on how they handle a request from end devices and how to act accordingly. They built a decision tree by listing parameters to help consumers decide which implementation is right for them by means of physical proximity, context awareness, power consumption, computation time, and logical proximity. Those researchers also argued to leverage the decision tree to build a guider system to literally guide the consumer with use cases and requirements to choose an appropriate edge computing implementation due to lack of standardization that edge computing suffers.

Akshay Jain and P. Singhal [11] introduced the definition, concept, characteristics, and application areas of fog computing. They also discussed its differences to cloud computing and alternative technologies. Extensive review is conducted to illustrate the CISCO IOx minicomputer platform combination of linux OS, used for processes computation, and CISCO network operating system, which is used for communication. They emphasized the capability of leveraging the possibility to run third party applications by platform users due to open source nature of linux OS and the SDK, and middleware availability.

Weisong Shi et al [5] discussed the concept of edge computing by analyzing why we started need edge computing. They began with several case studies like cloud offloading, smart home and city, and collaborative edge to further illustrate edge computing vision. Furthermore, challenges faced with edge computing implementations had been mentioned including programmability, naming, data abstraction, service management, privacy, and security, along with optimization metrics that are worth future research and study. The idea of collaborative edge benefited in healthcare scenario was considered to link patients and cloud both physically and logically in wide areas by distribution of processing applications through pharmacies, hospitals, insurances, and pharmaceuticals.

Junjie Zeng et al [13] studied the capabilities AI brings when applied to edge devices. A case of a face recognition system to preprocessing the image/video that visual camera takes was considered in order to reduce the computation that previously completed by cloud, thus reducing latency of result coming back. A complete system had been built to prove effectiveness of proposed framework using the cloud server, rk3288 development platform, and webcam. To enhance the effectiveness and to mitigate latency, a modification to recognition algorithms used need to be done in the future.

Most of the current traditional edge computing architectures do not deliver ultra-low latency for specific application in real-time fashion and suffer from peak time overloading issues.

Thus, Cheikh Saliou M. Babou et al. [14] proposed a three-tier home edge computing (HEC) architecture to reduce the previously mentioned issues. They used a home server as a gateway to connect various equipments inside apartment, i.e. Raspberry Pi devices. An edge server was also used to handle user queries unresolved by home server. All of these along with a central cloud work together as a kind of distribute processing of data between home and edge server. The effectiveness of that architecture was proved by using EdgeCloudSim simulation platform. They significantly reduced requests time from users by appropriately setting the distance between home and edge servers, and the number of mobile devices included in generation of requests process.

Cloud Offloading

Anas Amjad et al [15] developed a framework for dynamic tracking, monitoring, and orchestration of cloud resources in IoT. It is based on assumption that cloud OS support bidirectional resource sharing between Local Processing Platform (LPP) with heterogeneity nature devices. They also developed an algorithm for computation offloading to tackle the issue of processing capacity shortage with focus conducted on integrating the advancement of edge computing resource requirement schemes along with resource allocation schemes found in enterprise cloud. Depending on resource request type from LPP (edge device), the remote peer cloud provider check wether available resources is sufficient to to be allocated to that LPP or decide to let another pre-agreed LPP to introduce available resource to the requested LPP. For reducing cost, researchers suggested to use auction mode in such framework.

Kiryong Ha et al [12] demonstrated how real time cognitive assistance applications for mobile users can shape the future leveraging from current building blocks of technology like wearable computers, cloud computing resources, ubiquitous wireless networks, and cognitive algorithms that exceed humans' capabilities such as natural language processing and speech recognition. The made a comparison based on different studies on human user latency delays in different such recognition subjects along with relative such latencies of wearable devices. Then, the need for offloading cognitive algorithms to be run on local server edge was demonstrated. The role of cloudlets as edge server was simplified by presenting a three-tier hierarchy of mobile device-cloudlet-cloud. A platform for cognitive assistance was created to run on top of cloudlets that have control VM that is responsible for manipulating sensor stream from wearable device. A collection of cognitive VMs processes streams of data based on its specific interest. A user guidance VM was used to forward user assistance from cognitive VMs. The focus was on architectural need and implementation issues with considering user experience enhancements and guidance for future work.

Mahadev Satyanarayanan [16] shared the astounding and yet fruitful journey of computing timeline from mobile computing to cloud emerging and hence the mobile-cloud convergence, the second-class data center, and cloudlets. They left openness and cohesiveness of software interfaces and network protocols for future consideration. The article argued for mobile devices cyber foraging to extend wireless nodes capabilities using wired hardware infrastructure and hence the cloud formed and founded. To this end, a more convergence between mobile devices and cloud began to be a primary discussion. Since then the proximity became be a necessity to serve some of hungry-resources and cognitive applications. That necessity was the driving force behind the emerging of Cloudlets.

By using Markov Decision Process (MDP), Khalid R. Alasmari et al [17] discussed whether to process specific contents of a mobile device application in the device itself, on edge, or in cloud. Such a choice could be made based on balance between computation time constraints and energy consumption issues in said mobile device. Their results showed savings of 17.47% to nearly 46.27% of energy consumption and execution time.

Since computation offloading had become a necessity to overcome cloud computing latency and bandwidth issues, a work done by Zhenguang Yu et al [18] tried to enable application/resource offloading leveraging containers technology comprised into architecture named Boundless Resource Orchestrator (BRO). They needed to schedule the offloaded application between cloud and edge (and among edge devices) in order to avoid bandwidth and computation congestion and hence increased latency. Thus, to reach the best resource utilization and/or better performance, they used an orchestration strategy named Best Performance at Least Cost (BPLC) to redirect application/resource among edge nodes. However, more enhancement in performance regarding containers and orchestration technologies need to be put in near future consideration.

REPORTING ON OUR ONGOING EDGE COMPUTING PROJECT

There are many challenging and open research areas in the emerging field of edge computing and IoT. These areas cover a wide spectrum including communications between the edge and the cloud, communications between edge nodes, the need for parallel computation algorithm for optimization, preserving end user privacy, and security [19]. The main aim of our ongoing project is to develop an experimental edge computing platform. Various design parameters, optimization issues, and required algorithms need to be considered. In the first phase, the emphasis is on the issue of efficient cloud offloading. This issue needs to be addressed considering energy consumption under different scenarios. Another important parameter to be considered is the reliability of available internet links and their consequences.

In the second phase, the privacy and security issues need to be carefully studied. The initial application is a smart home or a smart office. Two main technologies will be used in our implementation, which are the Microsoft Azure platform and Raspberry Pi edge devices. In the following subsections, the two technologies are considered in a more detail.

Microsoft Azure Platform

Azure is a complete cloud platform that can host various applications, streamline the development of new applications, and also improve locally hosted or on-premises applications [20]. Azure Portal let developers access and maintain all azure services through the provided web interface.

Azure Hosting Services

Azure introduces several cloud-based computing resources to serve apps without user intervention need in terms of infrastructure maintenance. Different services are provided by Azure. A category of them is IaaS which gives developers full control over application hosting. Another Azure services like PaaS delivers fully managed services mandatory to support apps. In situations when the need only to write a code, this is supported by serverless hosting service [21].

Azure App service— serves primarily web-based apps, mobile backend apps, APIs apps and host them in Microsoft cloud. Azure app service is a container to different services like Wen apps, mobile apps, logic apps, API apps, and functions as a serverless event-based experience to accelerate developments.

Azure Virtual Machines— Azure virtual machines support the deployment of Windows or Linux VMs to Azure. Azure gives complete control of specification and configuration of created VM. The user is responsible for installing software, configuration, maintenance, and operating system patches because of her/his rule of providing IaaS. Virtual Machine are greatly useful for developers who need significant control over entire application infrastructure or locally hosted applications.

Azure Functions (serverless)— serverless coding is what meant by Azure Functions that let developers just write code and the execution of that code is triggered by HTTP request, in responds to other Azure Services, or at the schedule not depending on infrastructure or managing applications or other server-side matters.

Azure Enhancement Services

There are more of Azure services that provide enhancement to applications by hosting their data, containerization, monitoring, and authentication. These are described briefly in turn as following [20]:

Hosted storage and data access— Azure Cosmos DB, Azure Storage, and Azure SQL Database all provide storage of data in Azure while they are differentiated in types of data they store. The first service is optimal for applications that need documents, tables, or graph databases with multiple well-defined consistency models. While the second is for simple applications that require storage for tables, files, or such non-relational data. The last is a Microsoft SQL version adapted to run in cloud that provides relational data storage.

- Docker support: Docker containers, a form of OS virtualization, let developers deploy applications in more efficient and predictable way. A project code run on a local machine can be run in multiple heterogeneous machines by generating Docker image that contains the project code by means of Docker file. Then, containers run on other machines can run same project code from that Docker image [22].

- Authentication: authentication of the Microsoft multi-tenant, cloud-based identity and access management service is achieved by adding single-sign on (SSO) to the apps and integrating with Azure Active Directory (Azure AD). App Service Authentication provides built-in authentication to apps when using App Services to host them.
- Monitoring: by tracking apps usage, we maintain them from various issues. To monitor performance, we have to use several monitoring options. Visual Studio Application Insights, an Azure-hosted extensible analytics service that integrates with Visual Studio, can be used to monitor live web applications. Azure Monitor is a service that helps developers to visualize, query, route, archive, and act on the metrics and logs that are generated by Azure infrastructure and resources [20].

Azure IoT Hub

This is the most common fully managed service from Microsoft Azure that provides secure, bidirectional connectivity between huge number of IoT devices and cloud. Furthermore, several basic characteristics are presented in this type of Azure services such as device-to-cloud and cloud-to-device messaging capabilities, routing of telemetry data to other Azure services, query-ready storage for device metadata and synchronized state information, secure communications, and comprehensive monitoring for device availability [23].

In order to understand the role of Azure IoT hub in different IoT solutions, the devices are connected to a gateway, and this in turn can be in cloud or on-premises, through common protocol understandable by cloud. The gateway, or device directly, is connected to IoT hub and then to solution backend. There are on-premises or protocol gateways that are responsible of translating between cloud-understandable protocols such as MQTT, AMQP, and HTTPS [23]. The on-premises gateway provides more features such as the ability of doing data processing and analytics in the edge and transmits only sensitive data for further processing and storage in cloud. As a result, latency in IoT real time sensitive applications will be significantly reduced to mitigate the load on the network in terms of connectivity bandwidth. Figure 6 shows, from high level, the core subsystems together and directions of data flow of typical IoT solution.

Azure IoT Edge

Azure IoT edge is cloud service ready to implement Edge Computing scenarios by using on-premises gateway with modules inside it to process data and decide which sensitive data to be allowed to cross over to cloud. Indeed, it provides short-term storage capabilities to some case scenarios. Azure IoT Edge runtime, as depicted by Figure 7, is built from a number of modules that form a processing pipeline. Messages are flowing between these modules by existing communication medium, or broker [25].

IoT edge contains pre-built modules used as starting point for developers that perform fundamental gateway operations. Developers use an interface to construct their own specialized modules. The infrastructure is needed to deploy and run a combination of modules [25]. Messaging between modules can be obtained by using a broker. A module publishes messages to broker by using messaging patterns such as bus, or publish/subscribe. The responsibility of broker is to forward message to module connected to it [25]. For example, an efficient and simple Azure IoT edge solution can be built using a Raspberry Pi 3 as a gateway and a Bluetooth low energy (BLE) as end device sensor. The BLE sensor device collects temperature data and sends it to gateway, and then the data flows through modules. The first module may function as interface with end device. The next module performs end device MAC mapping to device id and key. The final IoT hub module sends the received data to IoT hub [26].

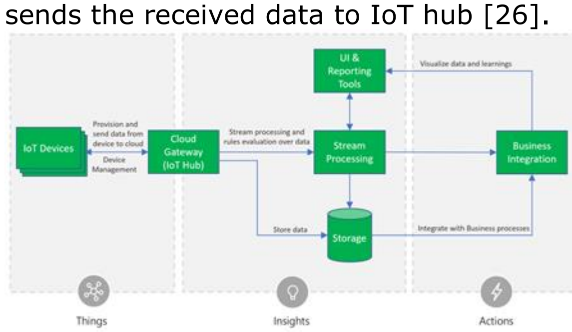

Figure 6.Core Subsystems of IoT Solution [24]

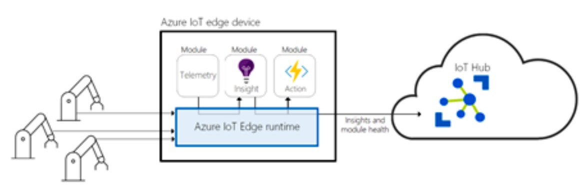

Figure 7.IoT Edge runtime [26]

RASPBERRY PI

Raspberry Pi is a small, affordable single board computer capable of doing what normal computer does. The reason for choosing this device because of its ability to run Linux Raspbian OS, Windows 10 IoT core, which is a special version of Windows targeting IoT. Also, it operates with popular visual studio IDE as well as its simplicity of development and deployment. Furthermore, it operates at low power and supports technologies that are open sourced. Raspberry pi is a suitable device for IoT and edge computing applications because of its characteristics among with its support for containers technologies [27].

Microsoft starter IoT pack for Raspberry pi [28] is a complete starter pack, Azure certified kit to be used with Azure IoT and to get started quickly. Because of light weight container technologies that are supported by Raspberry pi, edge computing scenarios are very possible to occur with this device. As an IoT enabled device, Azure IoT Edge is a cloud services to enable edge computing scenarios and implementation using the said device [29]. One of the simple use cases, which leveraging Azure Functions, IoT Edge service, along with this device to enable one of straightforward edge computing scenarios, is what Figure 8 shows. In this case, execution of deployed code is achieved inside Azure function on-premises to simplify triggering of events and process data on edge device near the source.

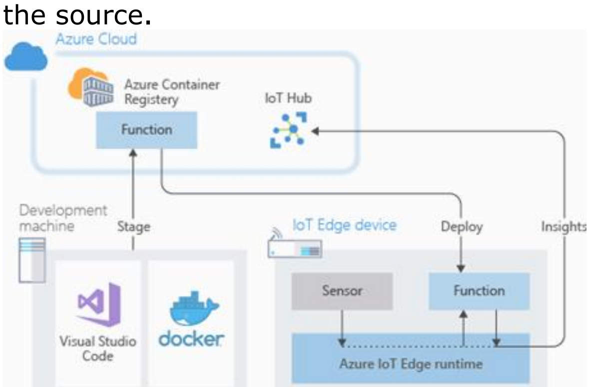

Figure 8.Deploying Azure functions as IoT Edge modules [30]

Conclusion

As the main contributions of edge computing is to offload computation near field devices. However, this is suffering from a number of challenges that researchers must take on their consideration to mitigate them gradually. One of these challenges that seem to be strange is lake of standardization in terms of compatibility between the protocols of devices used in implementation. Despite of these challenges, edge computing big beneficiary is related to wearable devices because of their resource constrained nature that make it necessary to offload the intensive processing to the nearest one-hop connectivity edge device. A lot of organizations are doing interesting work to make edge computing more seamless, useful, and also productive. Our ongoing project is a part of this global effort. Developing an efficient edge computing experimental setup would enable us to investigate various scenarios and optimization decisions to reach the best possible trade-offs in terms of different related edge computing parameters.

References

[1] M. Satyanarayanan, " The Emergence of Edge Computing," IEEE Computer Society, vol. 50, no. 1, 2017, pp. 30-39.

[2] William Stallings, Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, Pearson Education, USA, 2016.

[3] Hind Bangui, Said Rakrak, Said Raghay, and Barbora Buhnova, "Moving to the Edge-Cloud-of-Things: Recent Advances and Future Research Directions," Electronics, Vol.7, No. 309, 2018; doi:10.3390/electronics7110309.

[4] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," 2017 Global Internet of Things Summit (GIoTS), Geneva, 2017, pp. 1-6.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct. 2016.

[6] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet," ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 16, no. 3, pp. 2–13, 2012.

[7] Kusumlata Jain and Smaranika Mohapatra, Taxonomy of Edge Computing: Challenges, Opportunities, and Data Reduction Methods, in: Fadi Al-Turjman (Eds.): Edge Computing From Hype to Reality, EAI/Springer Innovations in Communication and

Computing, Springer Nature Switzerland AG 2019, Gewerbestrasse 11, 6330 Cham, Switzerland, pp. 51–71. https://doi.org/10.1007/978-3-319-99061-3

[8] Osamah Ibrahem Khalaf, Ghaidaa Muttasher et al., "Improving video Transmission Over Heterogeneous Network by Using ARQ and FEC Error Correction Algorithm", vol. 30, no.8, pp.24-27, Nov 2015

[9] Zhuo Chen, Lu Jiang, Wenlu Hu, Kiryong Ha, Brandon Amos, Padmanabhan Pillai, Alex Hauptmann, and Mahadev Satyanarayanan. 2015. Early Implementation Experience with Wearable Cognitive Assistance Applications. In Proceedings of the 2015 workshop on Wearable Systems and Applications (WearSys '15). ACM, New York, NY, USA, 33-38.

[10] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," 6th International Conference on Mobile Computing, Applications and Services, Austin, TX, 2014, pp. 1-9.

[11] A. Jain and P. Singhal, "Fog computing: Driving force behind the emergence of edge computing," 2016 International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, 2016, pp. 294-297.

[12] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2014. Towards wearable cognitive assistance. In Proceedings of the 12th annual international conference on Mobile systems, applications, and services (MobiSys '14). ACM, New York, NY, USA, 68-81. DOI=http://dx.doi.org/10.1145/2594368.2594383

[13] Junjie Zeng, Cheng Li, Liang-Jie Zhang, A Face Recognition System Based on Cloud Computing and AI Edge for IOT, in: S. Liu et al. (Eds.): EDGE 2018, LNCS 10973, Lecture Notes in Computer Science, Springer International Publishing AG, part of Springer Nature 2018, Seattle, WA, USA, June 25–30, 2018, pp. 91–98. https://doi.org/10.1007/978-3-319-94340-4_7

[14] Cheikh Saliou Mbacke Babou, Doudou Fall, Shigeru Kashihara, Ibrahima Niang, Youki Kadobayashi, Home Edge Computing (HEC): Design of a New Edge Computing Technology for Achieving Ultra-Low Latency, in: S. Liu et al. (Eds.): EDGE 2018, LNCS 10973, Lecture Notes in Computer Science, Springer International Publishing AG, part of Springer Nature 2018, Seattle, WA, USA, June 25–30, 2018, pp. 3–17. https://doi.org/10.1007/978-3-319-94340-4_1

[15] A. Amjad, F. Rabby, S. Sadia, M. Patwary and E. Benkhelifa, "Cognitive Edge Computing based resource allocation framework for Internet of Things," 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), Valencia, 2017, pp. 194-200

[16] Mahadev Satyanarayanan, A Brief History of Cloud Offload: A Personal Journey from Odyssey Through Cyber Foraging to Cloudlets. GetMobile: Mobile Comp. and Comm. 18, 4 (January 2015), 19-23. DOI: https://doi.org/10.1145/2721914.2721921

[17] Khalid R. Alasmari, Robert C. Green II, Mansoor Alam, Mobile Edge Offloading Using Markov Decision Processes, in: S. Liu et al. (Eds.): EDGE 2018, LNCS 10973, Lecture Notes in Computer Science, Springer International Publishing AG, part of Springer Nature 2018, Seattle, WA, USA, June 25–30, 2018, pp. 80–90. https://doi.org/10.1007/978-3-319-94340-4_6

[18] Ogudo, K.A.; Muwawa Jean Nestor, D.; Ibrahim Khalaf, O.; Daei Kasmaei, H. A Device Performance and Data Analytics Concept for Smartphones' IoT Services and Machine-Type Communication in Cellular Networks. *Symmetry* **2019**, *11*, 593.

[19] Hany F. Atlam, Robert J.Walters, and Gary B. Wills, "Fog Computing and the Internet of Things: A Review," Big Data and Cognitive Computing, vol. 2, no. 10, 2018; doi:10.3390/bdcc2020010.

[20] Getting Started Guide for Azure Developers. https://docs.microsoft.com/en-

us/azure/guides/developer/azure-developer-guide, 2019 (accessed 1 March 2019).

[21] Deploying Web Apps to App Service: Introduction. https://youtu.be/jCfJWYocz1M, 2019 (accessed 1 March 2019).

[22] GHAIDA MUTTASHAR ABDULSAHIB and OSAMAH IBRAHIM KHALAF, 2018. AN IMPROVED ALGORITHM TO FIRE DETECTION IN FOREST BY USING WIRELESS SENSOR NETWORKS.International Journal of Civil Engineering & Technology (IJCIET) - Scopus Indexed. Volume:9, Issue:11, Pages:369-377.

[23] Overview of the Azure IoT Hub service. https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-what-is-iot-hub, 2019 (accessed 3 March 2019).

[24] Ayman Dawood Salman1, Osamah Ibrahim Khalaf and Ghaida Muttashar Abdulsahib, 2019. An adaptive intelligent alarm system for wireless sensor network. Indonesian Journal of Electrical Engineering and Computer Science, Vol. 15, No. 1, July 2019, pp. 142~147

[25] Microsoft Azure documentations, What is Azure IoT Edge. https://docs.microsoft.com/en-us/azure/iot-edge/about-iot-edge, 2019 (accessed 1 March 2019).

[26] Forward device-to-cloud messages to IoT Hub using Azure IoT Edge on a Raspberry Pi. https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-iot-edge-physical-device, 2019 (accessed 1 March 2019).

[27] Osamah Ibrahim Khalaf, Bayan Mahdi Sabbar''An overview on wireless sensor networks and finding optimal location of node'',Periodicals of Engineering and Natural Sciences, Vol 7, No 3 (2019)

[28] Microsoft Starter Pack for Raspberry Pi. https://www.adafruit.com/product/2733, 2019 (accessed 1 March 2019).

[29] Azure IoT Edge service page. https://azure.microsoft.com/en-us/services/iot-edge/, 2019 (accessed 1 March 2019).

[30] Osamah Ibrahim Khalaf, Ghaida Muttashar Abdulsahib and Muayed Sadik, 2018. A Modified Algorithm for Improving Lifetime WSN. Journal of Engineering and Applied Sciences, 13: 9277-9282