IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Introduction to Edge Computing, Tools, and Challenges

Sanket Salankimatt[1], Prof. Anitha Sandeep[2], Dr. Vageesha S Mathada[3]

[1]8th Semester Student, Computer Science Department, RVCE, Bengaluru
[2]Assistant Professor, Computer Science Department, RVCE, Bengaluru
[3]Professor, Bheemanna Khandre Institute of Technology, Bhalki

Abstract - *The edge computing systems, powered by the dreams of the Internet of Things and 5G communications, combine computing, storage, and network services at the edge of the network to provide computing resources, allowing developers to rapidly create and implement edge applications. The edge computing systems have currently received extensive attention in various fields. This paper offers a detailed overview of the current edge computing devices and proposes representative projects in order to identify new research possibilities and assist users in choosing appropriate edge computing systems for specific applications. It provides a study of open-source resources according to their scope. Finally, we illustrate edge computing systems for energy conservation and deep learning optimization. This paper also looks at relevant problems for understanding and developing an edge computing framework.*

***Key Words*: Edge computing technologies, energy management, open source edge computing tools.**

1.INTRODUCTION

Throughout the post-Cloud age, Internet of Things (IoT) proliferation and 4G/5G commercialization slowly changed the population's practice of accessing and storing data, and questioned the monotonically growing ability of cloud computing. Edge processing is a modern computing model, with data being processed at the network edge. Promoted by the rapidly increasing demand and development in this field, the edge computing systems and technologies are flourishing, although some of them may not currently be popularly used. There are several viewpoints on differentiation in identifying various edge computing systems. In particular, based on specific design criteria, existing edge computing systems can be loosely be divided into three groups, putting together developments in system architecture, software models, and various implementations.

1) Push From Cloud: Cloud providers drive infrastructure and computing to the edge in this category to optimize locality, minimize response time, and enhance customer experience. Cloudlet, Cachier, AirBox and CloudPath are Component Systems. Most conventional cloud computing suppliers deliberately move cloud services closer to consumers, narrowing the gap between customers and cloud computing, so that mobile edge computing (MEC) does not lose its share.
2) Pull From IoT: To manage the huge amount of data produced by IoT devices, IoT applications pull resources and computing from the distant cloud to the close edges. s PCloud, ParaDrop, FocusStack, and SpanEdge are descriptive structures. Using edge computing systems and software, IoT products can share computing, storage, and bandwidth efficiently while retaining some degree of autonomy.
3) Hybrid Cloud-Edge AnalyticsIntegrating cloud and edge benefits offers a way to promote optimum global outcomes as well as reduced response time in sophisticated, digital services and applications. Representative systems involve computer systems including Firework and Cloud-Sea. These edge computing systems use IoT devices 'processing power to sort, preprocess, and integrate IoT data while utilizing cloud providers' capacity and versatility to perform complex analytics on those data.

Representative systems include Firework and Cloud-Sea computing systems. Such edge computing systems use the processing power of IoT devices to organize, modularize, and merge IoT data while making use of the ability and functionality of cloud providers to perform complex analysis on such data.

2. EDGE COMPUTING BASICS

Definition: Edge computing is a modern framework in which an edge server's services are located at the fringe of the Internet, near to mobile platforms, detectors, end-users and evolving IoT. "Edge" is characterized as any computer and network tools along the path between sources of data and cloud storage centers, and edge is a gradient, as seen in Figure 1.

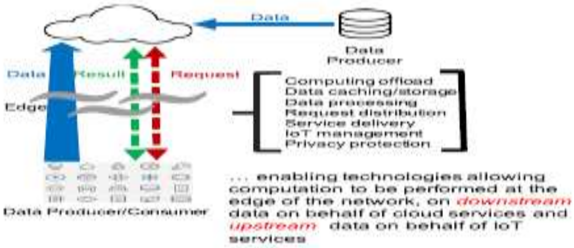

Figure 1. Edge computing in brief

2.1 3 Tier Edge Computing Platform

Analyzing several typical edge computing technology examples, in Fig. 2, we describe a standard model of three-tiered edge computing: IoT, edge, and cloud. Its first tier is IoT, including drones, connected health area cameras, smart home devices and appliances, and Industrial Internet equipment. To link IoT and the edge, several communication protocols are used. For example, drones can connect to a 4G / LTE cellular tower, and sensors can communicate with the home gateway through WiFi in smart home. Edge, like linked and automated vehicles, mobile tower, gateway, and edge servers, includes the cloud's immense processing and storage capacities to complete challenging tasks. The protocols here between IoT and the edge typically have lower power consumption and small distance characteristics, whereas the protocols here between edge and cloud have wide throughput and high speeds. The chosen communication protocols for edge and cloud are the Ethernet, optical fibers, and the upcoming 5G.

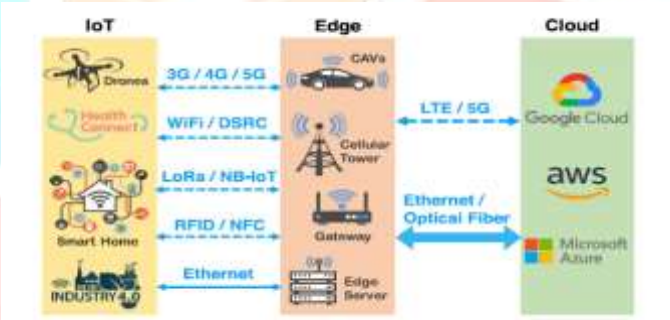

Figure 2- 3 tier Edge Computing Platform

3.EDGE COMPUTING PLATFORMS AND TOOLS

3.1 Cloudlet

Cloudlet is a secure, resource-rich computer or computer cluster that is very well-connected to the Internet and accessible for mobile devices in the vicinity. The cloudlet is in the middle layer of the three-tiered edge computing framework and can be deployed on a personal computer, minimal-cost server, or tiny cluster. As the Cloudlet is just one hop away from the mobile devices of the users, it increases the QoS with low contact latency and higher bandwidth utilization.

Cloudlet has 3 primary features given below:

1) **Soft State**: Cloudlet can be called a micro cloud computing center situated at the network's edge. Unlike Cloud, however, Cloudlet does not retain long-term contact status information but only temporarily stores certain state information.
2) **Rich Resources:** Cloudlet has enough computing resources to allow growing mobile users to discharge computing tasks into it.
3) **Close to Users:** Cloudlets are installed where both network range and physical distance to the end user are short, making it possible to monitor the bandwidth, latency and jitter of the network

As shown in Figure 3, Cloudlet mobility enables applications, based on three main measures

1) **Cloudlet Discovery:** Mobile devices are able to quickly discover the available Cloudlets towards them and pick the most appropriate one for downloading tasks.
2) **Virtual Machine (VM) Provisioning:** Configure and deploy the software VM containing the application code on the cloudlet so it is ready for the client to use.
3) **VM Handoff:** Migration to another cloudlet through the VM running the program.

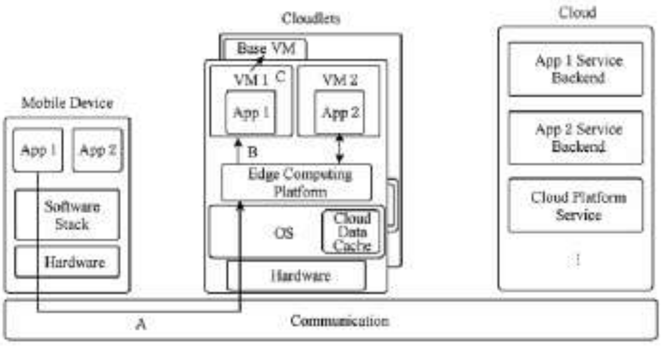

Figure 3. An summary of the cloudlet components and functions that enable versatility of applications. A: The discovery of cloudlets. B: The provisioning of VM. C: A handover of VM

3.2 CloudPath

Diverse resources, such as processing and storage, are given in such a framework along the path from the user computer to the cloud. It facilitates on-demand allocation and multilevel design dynamic deployment. CloudPath's key concept is to incorporate the so-called "path computing," so that compared to traditional cloud computing it can reduce the response time and increase bandwidth utilization. As seen in Fig. 4, CloudPath's bottom layer is consumer computers, and the top layer is cloud computing DC. The framework reassigns certain DCs tasks along the path (for path computing) to support various forms of applications like IoT data collection, data storage services, and data analysis services. Path computing creates a multi-tiered structure, but from the top (traditional DC) to the bottom (user terminal equipment) the computer capability becomes weaker, whereas the number of users becomes greater. The CloudPath application comprises of a collection of short-cycle and stateless functions which can be deployed easily at any level within the CloudPath system. Developers often tag functions to determine where their codes are running (e.g. edges, cores, clouds, etc.) or tag performance criteria (e.g. response latency) to approximate the running location. CloudPath does not relocate a running function/module but, by stopping the current version and restarting a new one at the planned spot, it enables device mobility.

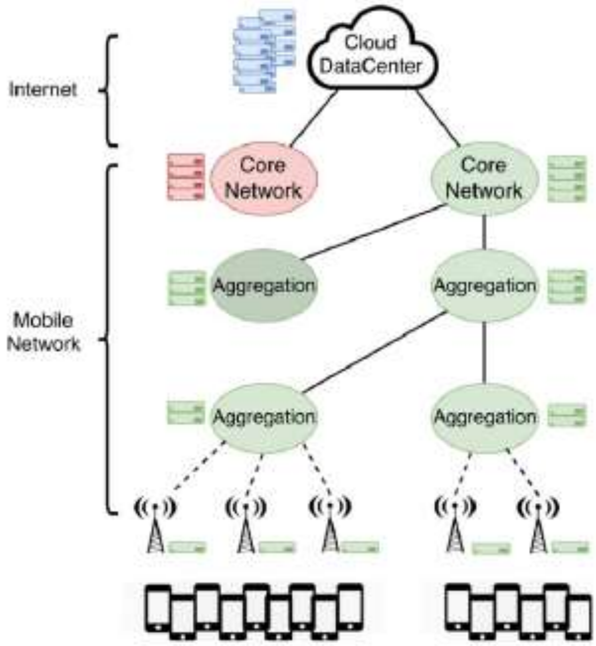

Figure 4- CloudPath Structure

4. OPEN-SOURCE EDGE COMPUTING PROJECTS

4.1 AKRAINO EDGE STACK

Akraino Edge Stack, which was started by AT&T and is now managed by the Linux Foundation, is a project to develop a holistic edge architecture solution to support edge cloud services with high availability**.** To have a holistic solution, Akraino Edge Stack has a broad range from the layer of infrastructure to the layer of application. Fig. 5 Reveals three-layered design. Akraino Edge Stack aims to build an environment with a virtual network function (VNF) in the application layer, and calls for edge applications. The second layer is middleware which supports top layer applications. Akraino Edge Stack aims to build Edge API and interoperability system within this layer with third-party Edge projects such as EdgeX Foundry. At the ground layer Akraino Edge Stack aims in cooperation with upstream communities to build an open-source software platform for the edge network. It interfaces and max imitates the use of current o pen-source projects like Kubernetes, OpenStack, etc.

Figure 5. Akraino Edge Stack Array

4.2 AZURE IOT STACK

Azure IoT Edge, which is supported as a cloud service provider by Microsoft Azure, is attempting to push cloud analytics to the edge applications. These edge devices could be routers, gateways, or other devices that can provide computational power. Azure IoT Edge's development model is like other cloud-based Azure IoT services that allow the user to transfer their current apps from Azure to the edge devices for lower delay. Azure IoT Edge consists of three components: IoT Edge units, IoT Edge running time, and a cloud-based framework as shown in Fig. 6. The first two components operate on edge computers, and the third is a cloud-based system. The first two elements operate on edge computers, and the last is a cloud-based system. IoT Edge nodes are containerized objects that run the application code or Azure applications. Such modules are controlled by IoT Edge runtime. The cloud-based framework is used to track and manage the previous two elements, meaning edge devices are monitored and managed. IoT Edge modules are the places where the processing units operate different applications. An image of a module is a Docker image which contains the client code.

IoT Edge runtime on the edge tools serves as boss. There are two modules: IoT Edge core and IoT Edge handler. IoT Edge hub acts as a virtual proxy for the cloud-based IoT Hub, which is a managed infrastructure and a central communication center. IoT Edge platform as a message broker lets modules connect with each other and transfer data to the IoT Platform. IoT Edge Agent is used to install and track modules for IoT Edge. This collects the application configuration information from the IoT Hub, initializes these modules and guarantees that they are operating and reboots crashed modules . Furthermore, it reports the module status to the IoT hub. Edge cloud framework is used for device management. Users may build edge applications via this interface, then submit those applications to the device, and eventually control the device's running status. This control feature is useful for use of huge systems, where consumers can deploy apps to large-scale machines and monitor them.

A basic platform delivery protocol is: Users select an Azure framework or write their own code as a platform, create it as an IoT Edge module file, and use the IoT Edge interface to deploy this module file to the edge unit. The IoT Edge then receives the configuration data, pulls the image of the node, and initializes instance of the node.

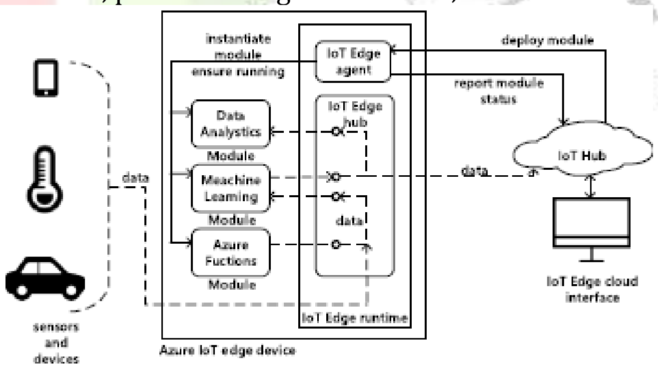

Figure 6. Azure IoT edge diagram

5. INCREASING ENERGY EFFICIENCY OF EDGE COMPUTING DEVICES

Energy consumption is often considered one of the big concerns in the design of an edge computing device, and is measured as one of the primary performance measures. In this section, we analyze the energy-efficiency-enhancing strategies implemented by state-of-the-art edge computing systems, respectively from the point of view of the top cloud layer, middle edge server layer, and bottom device layer.

5.1 Top Cloud Layer

A centralized dc can take hundreds of servers for cloud storage and therefore consume tremendous power. Will the edge / fog computing model need more or less power as an alternative to cloud computing? Multiple points of view have been expressed: some argue that distributed data storage and processing enabled by the edge computing architecture are more energy efficient while others prove that distributed information can absorb more resources than centralized content distribution.

The findings suggest that the fog mode can be of greater energy efficiency, based on many device architecture considerations (e.g. type of program, type of access network, and active time ratio), and certain systems that produce and transmit vast volumes of data in end-user premises result in the best energy savings under the fog mode.

5.2 Middle Edge Server Layer

Energy is often seen as an significant factor of the middle layer of the edge computing model, as the edge servers can be installed in a domestic environment or operated by the pump. Thus, in order to have a higher accessibility, many power management strategies were introduced to restrict edge server power consumption while also maintaining their efficiency. We provide an overview of two main techniques used in recent edge computing systems on the edge server layer.

1) *Design and power control of low-power devices:*
   A system-oriented fog / edge computing system, Fog Information, is developed and tested in. It is introduced with an integrated computer framework and the wearable sensors conduct data collection and data processing on the raw data collected. For transmission, orders of magnitude data are popular with Fog Data, which contributes to significant energy savings. In comparison, Fog Data is built for lower power architecture, which uses much less power than a Raspberry Pi.
2) *Sustainable Green Edge Powered Computing*:
   Multiple energy supplies are used to support the operation of a network based on fog computing, where solar power is used as the main energy supply of the fog nodes. A rigorous theoretical structure is introduced to simulate the long-term cost of consuming energy. Meanwhile, the architecture also makes a system for power-efficient data offloading to help provide high quality of operation.

5.3 Bottom device layer

The IoT systems in edge computting typically have stringent energy restrictions like a well-recognized reality, e.g. reduced battery life and power storage. So, powering a large number of IoT devices at the edge, particularly for those resource-intensive apps or services, remains a main challenge.

6. KEY DESIGN CHALLENGES

The edge computing network handles diverse services along the road from the cloud center to end users, protects device complexity and variety, and helps developers build and implement new technologies easily. In order to better maximize the benefits, we address the following main problems that need to be taken into account when evaluating and developing a modern edge computing system.

1) *Supporting Mobility*: Aid for mobility has two facets of it: versatility of consumers and stability of services. User agility applies to how the existing system state and required data can be dynamically transferred as the customer transfers from one edge node coverage area to another, so that the functionality is smooth. Capacity agility applies to: 1) how to actively identify and handle available tools, both long-term and short-term tools, and 2) how to restart the network as quickly as possible with substitutes as such edge devices are aged.
2) *Protection of Privacy*: Unlike cloud storage, edge devices such as portal tools for smart home applications, may be privately owned. When other people use these edge tools, get their data and even take ownership of them, it is necessary to protect the privacy of the owner and the data protection of the guest people. Improving network separation, developing privilege management and access control policies can be possible avenues for addressing the privacy problem.
3) *Friendliness to the application developer*:
   Ultimately, the device provides the relationship between hardware and basic resources for applications at the top level. The key considerations for a broad use of the framework are how to build responsive APIs, software delivery module, tool configuration and cancellation, and so on. To design an edge computing device we will also consider from the viewpoint of an application developer. In particular, delivering efficient engineering and implementation tools is a smart strategy to help the edge computing ecosystem grow.
4) *Compatibility*: There are actually quite a few advanced edge computing implementations. Popular programs to run in edge systems are not explicitly supported. Whether to migrate current systems to the edge version seamlessly and transparently to leverage the benefits of edge computing are still open questions.

CONCLUSIONS

Edge computing is a new revolution which jointly relocates networking, computation, and storage capabilities from the distant cloud to the client sides. In the sense of IoT and 5 G, edge computing's dream is exciting in delivering improved user interfaces for smarter networks and applications. The newly developed edge computing systems and technologies typically minimize overhead data collection and storage, and increase the reliability and effectiveness of mobile data analytics. This paper presented the representative structures and open-source edge computing initiatives, addressed some energy-efficiency-enhancing approaches for analysis of results

REFERENCES

[1] (2016). *Introduction of Edge Computing Consortium*. [Online]. Available: http://en.ecconsortium.org/

[2] M. Chiang and W. Shi, "NSF workshop report on grand challenges in edge computing," Tech. Rep., Oct.2016.[Online]. Available:http://iot.eng.wayne.edu/edge/NSF%20Edge%20Workshop%20Report.pdf

[3] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, ETSI, White Paper 11, 2015, pp. 1–16, vol. 11, no. 11.

[4] R. LaMothe, "Edge computing," Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep., 2013. Accessed: Mar. 2014. [Online]. Available: https://mafiadoc.com/edge-computing-pacificnorthwest-national-laboratory_59d648481723dd08e35b7b77.html

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

BIOGRAPHIES

Sanket Salankimatt is currently pursuing his B.E. Degree in Computer Science Engineering from R.V. College of Engineering, Bengaluru. His research interests include Cloud Computing, Data Science and Machine Learning.

Prof. Anitha Sandeep is working as Assistant Professor in Computer Science Department at RVCE, Bengaluru. Her research interests include Game Theory, Fuzzy Logic and Graphs, and Theoretical Computer Science.

Dr. V S. Mathada is working as Professor at BKIT, Bhalki. He has around 30 years of teaching, professional and research experience in India and abroad. His research interests are Reliability Engineering, Fuzzy Techniques, etc.