Advances in Social Science, Education and Humanities Research, volume 577
Proceedings of the 2021 International Conference on Modern Management and Education Research (MMER 2021)

Comparative Research on Edge Computing System

Jiedong Bi*, Xinchang Zhang, Wenpeng Cao

Qilu University of Technology, Shandong Computer Science Center, Shandong, China
**Corresponding author. Email:1113052245@qq.com*

ABSTRACT
Edge computing transfers computing tasks from centralized management to the edge of the network, providing services nearby, reducing the amount of data transmission and ensuring data security. In order to study and apply edge computing system, this paper mainly discusses the background and current situation of edge computing. It introduces several typical edge computing systems, and compares them in the aspects of system characteristics, application fields and limitations, so as to help users understand edge computing system.

Keywords: *Edge computing system, System characteristics, Application fields, Limitations.*

1. INTRODUCTION

Edge computing technology is a new distributed computing model that provides real-time data computing services by extending computing tasks from the centralized cloud to the edge near IoT devices and thus. In recent years, with the rapid development of Internet of things and edge computing technology, there are many edge computing systems in different application levels, such as EdgeX Foundry, Link IoT Edge, Apache Edgent, AWS IOT Greengrass, Akraino Edge Stack, etc.

2. EDGE COMPUTING

Since the edge computing was proposed by Garcia et al [1] in 2015, the edge computing technology has developed by leaps and bounds. Edge computing is a decentralized computing model that migrates data, applications, or services from the central node of the network to the logical edge nodes of the network for processing. Edge computing is characterized by high bandwidth, ultra-low latency and real-time access to network information, which can be used by a variety of applications [2]. Edge computing has a wide range of applications in the Internet of things, such as smart city, health care and traffic management.

3. EDGE COMPUTING SYSTEMS

Currently, edge computing systems are still in their infancy. There are various design solutions of each edge computing system for different problems. Currently, according to the design objectives and deployment methods of edge computing systems, it can be divided into three types of edge computing systems: IoT-oriented, edge cloud services-oriented, and cloud-edge convergence-oriented.

The edge computing system for IoT end is dedicated to solve the problems brought by design, development and deployment during IoT application. IoT requires fast response rather than high computing power and large storage capacity. Currently, representative systems are EdgeX Foundry from the Linux Foundation and Apache Edgent from the Apache Software Foundation.

3.1.1. EdgeX Foundry

EdgeX Foundry is an open source project of the Linux Foundation, which aims to simplify and standardize the edge computing framework in the IoT market. The core of it is an interoperability framework based on the reference software platform which has nothing to do with hardware and operating system.

Figure 1 shows the architecture of EdgeX Foundry. EdgeX Foundry is a collection of open source micro services. These micro services are divided into four service layers and two base system service layers. As shown in Figure 1, the four service layers are the core service layer, the support service layer, the export service layer, and the device service layer. The two base system services are system management and security, respectively.

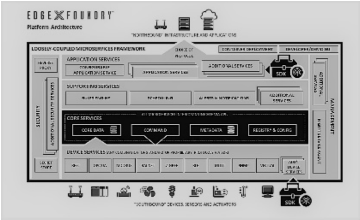

Figure 1 EdgeX Foundry architecture.

EdgeX Foundry's application scenario is the industrial Internet of things, which is mainly used for sensors or actuators and other devices, such as automatic chemical plants [3]. It provides a variety of development tools to help developers create edge solutions for the Internet of things based on EdgeX. At present, EdgeX Foundry is in the stage of rapid development, its user interface is being developed, and more functions will be added in the future version.

3.1.2. Apache Edgent

Apache Edgent is a microkernel-style programming model that runs primarily in resource-constrained edge computing nodes and can be embedded in gateways and small IoT devices. Apache Edgent can perform real-time analytics on data streams from edge devices, sensors, actuators, etc. It can be used by combining with centralized analytics systems and provides efficient and timely analytics and management capabilities for the entire IoT ecosystem.

Figure 2 illustrates the Apache Edgent application development model. Apache Edgent performs control loops through the central analytic device model. Apache Edgent sends device events for centralized analysis and receives device commands to change behavior. Apache Edgent device program analyzes the data from the edge application device, and analyzes the important data sent to the back-end system.

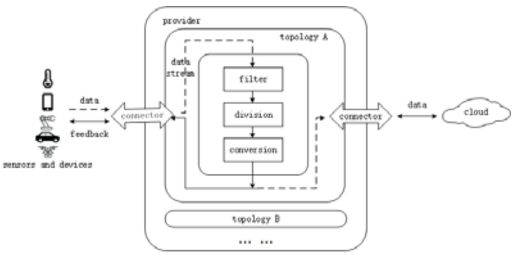

Figure 2 Apache Edgent application development model.

Apache Edgent is mainly used in the following scenarios: Real-time analysis of data on edge devices and mobile devices; Real-time analysis of application server error log; Real-time analysis of server health.

3.2. Edge Computing System for Edge Cloud Services

Edge cloud service solves the problem of Internet of things resource shortage by providing local computing and network resource storage. Representative systems are CORD from the Open Networking Foundation (ONF) and Akraino Edge Stack from the Linux Foundation.

3.2.1. CORD

CORD is a major project promoted by ONF, AT&T, and other carriers and vendors to build cost-effective and flexible access networks based on common hardware, open source software, software-defined networking (SDN) [4], network function virtualization (NFV) [5], and elastic cloud services [6], and to enable rapid response and flexible deployment of edge computing applications. CORD aims to provide network operators with the economic benefits of infrastructure construction and agility benefits such as rapid deployment and elastic scaling services [7].

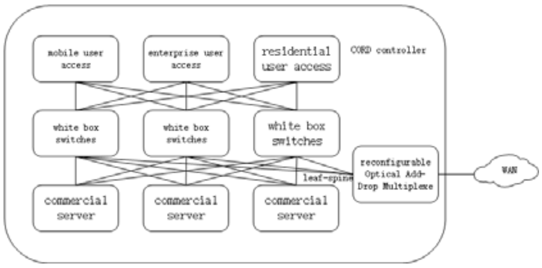

Figure 3 CORD hardware architecture.

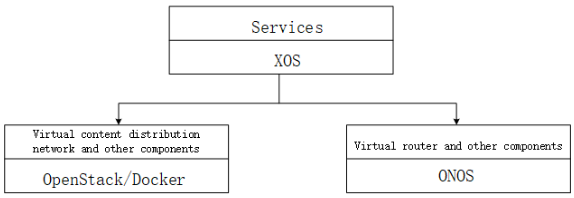

Figure 4 CORD software architecture.

CORD is a system composed of hardware and software. Figure 3 shows the hardware architecture of the CORD. CORD is mainly divided into three layers: user access layer, white box switch layer, and commercial server layer. It uses the commercial servers interconnected by white box switching structure to provide computing, storage, management and other functions. Its switch architecture adopts a lateral flow optimized ridge topology switch architecture, so it has scalable throughput. At the same time, it also has access hardware for connecting users to improve the real-time performance of user access. Figure 4 shows the software architecture of the CORD. OpenStack is a cluster management suite, which provides the core internet as a service function. Docker provides a container based approach to deploy and interconnect services. ONOS is a network operating system that manages the structure of software switch and physical switch. XOS is a composite service framework.

CORD replaces closed, proprietary hardware with software that runs on commercial servers, switches and access devices. It enables network operators to benefit from economies of scale and agility. By now, the deployment of CORD is still in the testing stage, so more research efforts need to be put into the combination of CORD and edge applications.

3.2.2. Akraino Edge Stack

Akraino Edge Stack is an open source project for high-performance edge cloud services and provides solutions for fully integrated edge infrastructures. Akraino Edge Stack is dedicated to the development of a set of open source software stacks that are primarily used to optimize the construction and management of edge network infrastructures to meet edge cloud services in terms of latency, performance, scalability, availability, and other requirements [8].

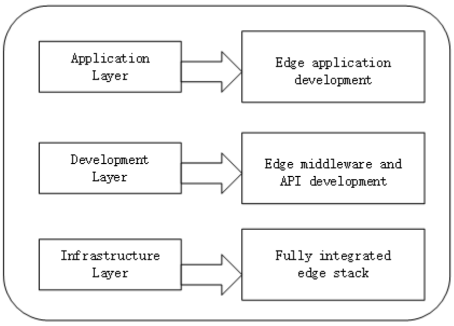

Figure 5 Akraino Edge Stack architecture scope.

The scope of the Akraino Edge Stack project can be divided into 3 levels ranging from infrastructure to edge computing applications as shown in Figure 5. At the top application level, Akraino Edge Stack is committed to building and promoting the development of edge computing applications. The middle tier focuses on middleware and API development to enable interoperability with third-party providers and hybrid edge cloud models. At the underlying infrastructure level, Akraino Edge Stack works with the upstream community to build a fully integrated edge blueprint for edge infrastructure optimization providing an open source software stack. In addition, Akraino Edge Stack provides configurations for each use case, including for hardware, software, management tools, and delivery points.

Akraino Edge Stack provides edge cloud services for edge applications and maximizes network transport efficiency and high availability of edge application services, and can be deployed in network operators or cable centers, etc., and has a wide range of application areas such as intelligent city, intelligent traffic, edge video processing.

3.3. Edge Computing System for Cloud-edge Convergence

Cloud servers have rich computing and storage capabilities, while edge computing has the advantages of low latency, high security, low bandwidth consumption, and context awareness, so combining cloud computing with edge computing, i.e., cloud edge computing framework, enables edge locations to form an elastic cloud platform with comprehensive set of computing, storage, security, resource management, and other capabilities. Representative systems are AWS IoT Greengrass from Amazon.com, Azure IoT Edge from Microsoft, and Link IoT Edge from AliCloud.

3.3.1. AWS IoT Greengrass

AWS IOT Greengrass can extend the cloud function to local devices, making devices have more efficient data collection and analysis capabilities when they are close to data sources. At the same time, it can create server less applications through lambda functions and connectors to quickly respond to local events, safely communicate with local resources, and efficiently process data on devices, so as to minimize the cost of data transmission from devices to the cloud [9].

Figure 6 demonstrates the basic architecture of AWS IoT Greengrass. Any device can interact with AWS IOT Greengrass core through local network using FreeRTOP or SDK configuration. It is responsible for authentication, authorization and secure message routing between locally deployed lambda functions and cloud through MQTT protocol [10]. In AWS IOT Greengrass, devices can safely communicate and exchange messages on the local network without having to connect to the cloud. AWS IOT Greengrass provides a local publish/subscribe message manager, which can intelligently buffer messages in case of loss of connection, so as to retain the inbound and outbound messages of the cloud.

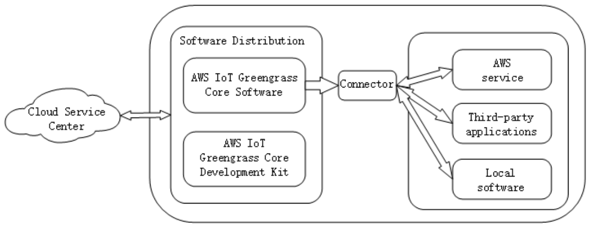

Figure 6 AWS IoT Greengrass basic architecture.

AWS IoT Greengrass offers benefits such as real-time response to local events, offline operation, and secure communication, while it can reduce the cost of running IoT applications by enabling simplified device programming with AWS Lambda functions.

3.3.2. Link IoT Edge

Link IoT Edge is an important project of AliCloud in the field of edge computing, which extends AliCloud's advantages in cloud computing, artificial intelligence, and big data closer to the edge of the network, creating a collaborative computing system that integrates "cloud, edge, and end".

The product architecture of Link IoT Edge mainly consists of three parts: the device side, the edge computing side, and the cloud side shown in Figure 7. Link IoT Edge is designed for IoT developers and inherits the capabilities of streaming computing, secure storage, intelligent management, machine learning, etc. It can be deployed in smart devices and edge computing nodes of different scales, making local computing services with the advantages of low latency, low cost, high security, high intelligence and easy scalability, while combining AliCloud's capabilities in big data, voice recognition, video processing, intelligent learning, etc., greatly improve the processing efficiency of local devices and reduce the load on the cloud.

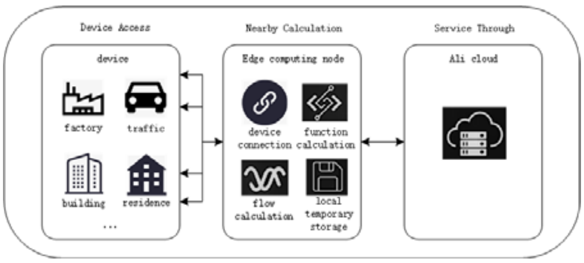

Figure 7 Link IoT Edge product architecture.

In practical scenarios, the application areas of Link IoT Edge are becoming more and more extensive. For example, in the hotel of the future, after the rapid integration of local devices through the edge gateway and as a local node to quickly respond to local events, the intelligent linkage of local devices such as water and electricity, air conditioning, lighting, room doors, etc., to achieve indoor and outdoor integration of voice intelligence. In wind power generation, by deploying edge computing gateways, local device data is collected in real time and uploaded to AliCloud for big data model training to provide timely feedback and optimization of power generation parameters, such as wind sensitivity and start-up delay parameters.

4. COMPARATIVE STUDY

In this section, we compare the performance of these edge computing systems in different aspects, as shown in Table 1, including the main applications, application areas, target users, system characteristics and limitations of the system.

In terms of main applications, EdgeX Foundry is mainly used to interact with sensors, actuators, devices and other IoT objects in the Internet of Things. Apache Edgent is used at the edge layer for real-time data stream processing. CORD mainly builds edge network infrastructure to improve the operation of edge network services and enhance the capabilities of the edge data center business cloud platform. Akraino Edge Stack improves the edge cloud infrastructure of enterprises, Internet service providers and operators to provide users with new flexible services. AWS IoT Greenras allows devices to operate on local data, and can use the cloud for computing, storage, management and analysis. Link IoT Edge provides control over millions of edge nodes to brings cloud capabilities down to the edge to provide computing services close to the user.

In terms of application areas, EdgeX Foundry interacts with IoT devices, sensors, actuators, etc. through an interoperability platform. Apache Edgent analyzes and processes real-time data generated by devices such as sensors and actuators of the Internet of Things. CORD and Akraino Edge Stack are both for enterprises, network operators and network service providers. AWS IoT Greengrass has a huge cloud service ecosystem that can provide manufacturers and users with efficient device data collection, storage, and analysis services. Link IoT Edge connects devices with different protocols and different data formats through defined object models to provide safe, reliable, low-latency, low-cost, easy-to-expand, and weakly dependent local computing services.

Table 1. Comparison of characteristics of edge computing systems

Edge Systems / Characteristics	EdgeX Foundry	Apache Edgent	CORD	Akraino Edge Stack	AWS IoT Greengrass	Link IoT Edge
Main Applications	Delivering Interoperability for the IoT Edge	Real-time analysis of continuous data streams	Service delivery platform for network operators	Improve the edge cloud infrastructure of enterprises, Internet service providers and operators' edge networks	Local computing, messaging, data management, analytics and persistent storage	Sink cloud capabilities to the edge side and deploy them in smart devices and computing nodes of different magnitudes
Application Areas	IoT	unlimited	unlimited	unlimited	unlimited	IoT

Target Users	users	users	Network Operators	Network Operators	Network Operators and users	Network Operators and users
System characteristics	General API interface for device manageme nt	Functional API for data analysis	It has all the built-in service functions required to fully operate the edge data center	Extensive edge cloud services	Real-time response to local events, offline operation, simplified device programming	Provides folder-like management capabilities, fast response to local events, offline operation, and fast programming
Limitations	Lack of programma ble interface	It can only be used for data analysis	Unable to run offline	Unable to run offline	No computing power provided	Unable to provide implementation of edge-to-cloud access solutions

In terms of target users, EdgeX Foundry is mainly deployed on edge devices. Apache Edgen runs in edge computing nodes with limited resources. CORD and Akraino Edge Stack are deployed on important edge infrastructure, and their targets are network operators, service providers, etc. Link IoT Edge targets users who are primarily IoT developers and it can be deployed in different volumes of smart devices and edge computing nodes to provide stable, secure and diverse network communication.

In terms of system characteristics, EdgeX Foundry provides a common API interface for the deployment and monitoring of edge applications. Apache Edgent performs real-time data flow analysis and processing in edge computing nodes with limited resources, and can interface with a variety of back-end services. CORD has powerful edge data center business cloud platform capabilities. Akraino Edge Stack has high-availability cloud services optimized for edge applications, which can quickly expand edge cloud services. AWS IOT Greengrass extends the cloud function to local edge devices, enabling devices to collect, store and analyze data close to information sources more effectively, and respond to local events independently, while enabling local edge devices to communicate with each other safely. Link IoT Edge provides a folder-like management capability for gateways, sub-devices, sensors, actuators, etc. at the edge, deploying data resources from edge instances to gateways, sensors, and other devices.

Finally in terms of limitations on deploying edge applications, EdgeX Foundry does not provide a programmable interface for applications written by developers. Apache Edgent is a lightweight system that is more focused on data analytics. AWS IoT Greengrass itself does not provide any computing power and is more of an orchestrator between devices provided by users outside of the AWS framework. Link IoT Edge has some usage limitations in terms of edge instances, drivers, and scenario linkage, while not providing an implementation of an edge-to-cloud access solution.

5. CONCLUSION

Today, more and more services are expanding from the cloud to the edge of the network, where data processing at the edge ensures shorter response times, lower network latency and load, and higher security. Currently, the development of edge computing systems is still in its infancy, and therefore, an in-depth understanding of them is of great research value. In this paper, we introduce six typical edge computing systems and conduct a comparative study of their characteristics to help users understand edge computing systems and make a choice.

REFERENCES

[1] P Garcia Lopez, A Montresor, D Epema, A Datta et al. "Edge-centric computing: Vision and challenges [C]." 2015, pp: 37-42.

[2] J Ren, Y He, G Huang et al. "An edge-computing based architecture for mobile augmented reality [J]." IEEE Network, 2019, 33(4): 162-169.

[3] JR Meredith. "Implementing the automated factory [J]." Journal of Manufacturing Systems, 1987, 6(1): 1-13.

[4] BBA Nunes, M Mendonca, XN Nguyen et al. "A survey of software-defined networking: Past, present, and future of programmable networks [J]." IEEE Communications surveys & tutorials, 2014, 16(3): 1617-1634

[5] H Hawilo, A Shami, M Mirahmadi et al. "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC) [J]." IEEE Network, 2014, 28(6): 18-26.

[6] S He, L Guo, Y Guo. "Real time elastic cloud management for limited resources [C]." In 2011 IEEE 4th International Conference on Cloud Computing, 2011, pp. 622-629.

[7] XR Lozano. "Deploying and testing of central office re-architected as a datacenter (CORD) [D]." Bachelor's thesis, Universitat Politècnica de Catalunya, 2018.

[8] F Liu, G Tang, Y Li et al. "A survey on edge computing systems and tools [J]." Proceedings of the IEEE, 2019, 107(8): 1537-1562.

[9] A Kurniawan. Learning AWS IoT: Effectively manage connected devices on the AWS cloud using services such as AWS Greengrass, AWS button, predictive analytics and machine learning [M]. Packt Publishing Ltd, 2018.

[10] SA Shinde, PA Nimkar, SP Singh, VD Salpe, YR Jadhav. "MQTT-message queuing telemetry transport protocol [J]." International Journal of Research, 2016, 3(3): 240-244.