Research on deployment method of edge computing gateway based on microservice architecture

Hai Zhang¹, Zhengyi Liu¹, Yaoyao Zhang¹, Shuguang Zhan¹, Jianjun Yang²,³, Guixiang Wang² and Yufei Sun²

¹Fushun Power Supply Company State Grid Liaoning Electric Power Supply Co,Ltd. Fushun, China;
²College of Information Science and Engineering Northeastern University, Shenyang, China
³Email: 945623803@qq.com

Abstract. Traditional solutions that use servers or computers to deploy on-site are bulky and costly. The edge computing system can be deployed in a lightweight and flexible manner according to the actual needs of the site, so as to achieve the lowest cost while meeting the functional requirements. Based on this, this article proposes an edge computing system solution based on microservice architecture, discusses the overall framework of the system and the implementation of microservice architecture in edge computing systems, and introduces in detail an edge computing gateway device deployment method based on the Gaussian mixture model, which can optimize the number of deployed gateway devices while ensuring the load balance of each gateway device. Finally, the advantages of the deployment method are verified through experiments.

1. Introduction

The data volume and data types of the ubiquitous power Internet of Things have increased dramatically[1], leading to the key to centralized big data center on the cloud computing model technology cannot efficiently process the massive data generated by edge devices, so the edge computing model came into being[2].

For edge computing, it is not only necessary to establish the gateway and other devices in the edge computing framework, but also to consider the deployment and positioning of gateway devices providing edge computing services[3]. How to reasonably distribute edge gateway devices to achieve the optimal network performance, so as to better complete edge computing services, this is a problem worth studying.

At present, many papers at home and abroad focus on the allocation and deployment of edge computing resources[4], however, in most of today's networks, terminal equipment is highly mobile[5-6]. If the traditional clustering deployment method is adopted, as the location of the mobile terminal changes, the algorithm must calculate the distance to each cluster center every time, so that the calculation amount is very large each time[7]. Compared with other clustering methods, the Gaussian mixture model[8] only calculates the probability of each sample node on each Gaussian model component. Each edge gateway device can be regarded as a Gaussian distribution component, and the deployment location of the edge gateway device is the center of each Gaussian distribution. In practice,
we can define the data model with any reasonably mixed distribution, so the Gaussian mixture model is more in line with practical meaning and more convenient to calculate than other algorithms.

Based on this, this article proposes an edge computing system solution based on microservice architecture[9]. This paper mainly introduces a method of edge computing gateway device deployment based on Gaussian mixture model[10]. This method combines the direct connection between the terminal equipment and the edge gateway equipment in the edge network and its own mobility characteristics, and uses a soft clustering method to locate and deploy the gateway, thereby making the deployment and positioning more accurate. At the same time, the deployment method can not only realize the optimization of the number of deployed gateway devices, but also ensure the load balance of each gateway device, which has high application value in production and life.

2. Establishment of edge computing gateway equipment

2.1. Overall architecture of edge computing system

The edge computing platform can provide edge computing core services, device services and output services with a microservice architecture, realize service management, business orchestration, and connection management. Through the interaction with service management, the sinking of edge computing on the terminal side and the gateway side is completed, and the application of edge computing in intelligent manufacturing can be quickly realized.

![Figure 1. Overall framework of edge computing][8]

The overall framework of edge computing is shown in Figure 1. The edge computing microservice architecture is mainly composed of four service layers and service management extensions.

1) Basic service layer. The basic service layer mainly realizes the unified management of general services and edge devices, realizes the virtualization of computing resources, storage resources, and network resources, and provides upper-layer microservices with distributed deployment capabilities related to specific devices and specific resources. Computing power is mainly virtualized through Docker technology [5].
(2) Equipment service layer. The equipment service layer mainly connects with devices in the south and core services in the north.

(3) Core service layer. The core service layer mainly includes four core functions: metadata management, registration and configuration management, command configuration management, and core data management. At the same time, it realizes the isolation of the bottom device, the conversion of the upper layer protocol, and provides southbound interfaces such as REST, OPC-UA, MODBUS TCP, BACNEY, BLE, MQTT, and SNMP to the basic service layer.

(4) Support service layer. The support service layer is composed of rule engine service, scheduling service, log service, and alarm notification service. It works on the core service layer, and edge computing processing modules and edge computing application modules are deployed in the support service layer.

(5) Output service layer. The output service layer mainly realizes the data interaction between the edge computing platform and the third-party system. The third-party system uses registration to realize the subscription of the required data. The output service layer pushes the data to the specified interface through data publishing and cloud publishing.

(6) Service management. Service management mainly realizes the unified management of resources in the basic service layer, core service layer, support service layer, and output service layer. It provides business process management, connection management, monitoring management, configuration management and other related functions in specific business scenarios. Different resources in the framework are coordinated and scheduled with microservices.

2.2. Edge computing under microservices
The realization of the entire microservices includes the use of the following components: service governance, gateway services, load balancing, fault tolerance protection, distributed configuration center, message bus, distributed service tracking, etc.

The component used by service governance is Spring Cloud Eureka, which implements service registration and service discovery, including service registry and service providers.

The component used by the gateway service is Spring Cloud Zuul. Its main function is to realize the maintenance of routing rules and service instances, as well as to realize signature verification and login verification through filters.

The component used in load balancing is Spring Cloud Ribbon, which is mainly used to achieve client load balancing. There are mainly random strategies, linear polling strategies, retry strategies, minimum concurrency strategies, available filtering strategies, response time weighting strategies, and regional weighting strategy, etc.

The component used for fault-tolerant protection is Spring Cloud Hystrix. After joining Hystrix, when the service fails, an error response is returned to the calling method through the fault monitoring of the circuit breaker.

The component used by the distributed configuration center is Spring Cloud Config, which is divided into two parts, server and client, to realize centralized management of configuration, dynamic adjustment of configuration, and automatic update of configuration.

The component used by the message bus is Spring Cloud Bus, which realizes the receiving and sending of messages by integrating Rabbit MQ.

The component used in distributed service tracking is Spring Cloud Sleuth, by integrating Logstash or Zipkin, fast tracking is realized to find the root cause of errors, and the performance bottleneck on each request link is monitored and analyzed.
3. Gateway equipment deployment

3.1. Model establishment

The deployment method of edge computing gateway equipment based on Gaussian mixture model[10] proposed in this chapter is mainly divided into three steps to complete the deployment and positioning of gateway equipment.

The first step: Use several Gaussian model components that obey the normal distribution to build a probability distribution model, and use the Gaussian model components to fit all points in the region, set up an objective function to solve the maximum probability value of the distribution.

The second step: After solving the maximum probability value of the probability distribution model by the method of obtaining the maximum likelihood estimation value, the load of each Gaussian component is used as an index to find a load-balanced clustering deployment method.

The third step: To optimize the clustering effect, it is necessary to introduce the Bayesian information criterion[7] as the criterion for selecting the number of Gaussian components, the purpose is to ensure the optimal number of Gaussian model components of the final model iteration. Before establishing the mathematical model of load balancing, by introducing the idea of Gaussian distribution, the distribution of sample nodes is regarded as a probability distribution model. The Gaussian distribution function is as formula (1):

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$  \hspace{1cm} (1)

The random variable $X$ in the Gaussian distribution corresponds to the position of the terminal device $x$, $\mu$ represents the center of the gateway device, and $\sigma^2$ represents the degree of concentration from the center. These $K$ Gaussian model components form a Gaussian mixture model[10]. In this way, the distribution problem of terminal devices in the network is regarded as a problem that the sample nodes obey the distribution of the Gaussian mixture model for subsequent solutions.

Through the probability function of a single Gaussian distribution, the total probability of a certain sample node $i$ belonging to the distribution can be described as the sum of the probability that the node belongs to each Gaussian component.

After obtaining the probability that each sample node belongs to the distribution, the probability of each sample node is multiplied to obtain the probability value of the entire distribution. The final solution is the maximum probability value of the distribution, and the objective function expression is written as formula (2):

$$\max \prod_{i=1}^{N} \sum_{j=1}^{K} \alpha_j \times f(x_i; \mu_j, \sigma_j)$$ \hspace{1cm} (2)

After the maximum value of the above formula is obtained, the probability value of each sample node subject to each Gaussian distribution is obtained by parameters in turn, and the specific clustering situation is judged by this probability value.

Before clustering, establish the following mathematical model:

$$B_j = \sum_{i=1}^{N} p_{ij} \times c_{ij} \times y_{ij}$$ \hspace{1cm} (3)

$B_j$ represents the total load of the Gaussian component $g_j$, and its practical meaning is the load of the gateway device; $p_{ij}$ represents the probability that node $x_i$ belongs to Gaussian component $g_j$, $c_{ij}$ represents the load required by the gateway device $g_j$ to process the request of the terminal device $x_i$, $y_{ij} = 1$ means that the terminal device $x_i$ is clustered to the gateway device $g_j$. The average load of the entire model can be expressed as:
In order to ensure the load balance of the entire Gaussian mixture model, the objective function is:

$$B = \frac{1}{K} \sum_{j=1}^{K} \sum_{i=1}^{N} p_{ij} \times c_{ij} \times y_{ij}$$  \hspace{1cm} (4)$$

$$\min \sqrt{\frac{1}{K-1} \sum_{j=1}^{K} (B_j - B)^2}$$  \hspace{1cm} (5)$$

$$\sum_{i=1}^{N} y_{ij} \times c_{ij} < C_j$$  \hspace{1cm} (6)$$

Formula (6) indicates that the total workload of the devices within the coverage of the gateway device cannot exceed its own maximum load.

3.2. Determination of model components
The BIC criterion[10] is used to determine whether the parameter selection is reasonable. According to the Gaussian mixture model in this article, assume that $K$ is the number of free parameters to be estimated, that is, the number of Gaussian models, the terminal device node $X = \{x_1, x_2, x_3, \cdots x_n\}$ is the sample, and $N$ is the length of the sample. The expression of the corresponding BIC criterion is:

$$BIC = -2 \times \max \sum_{i=1}^{N} \log(p(x_i; \mu, \sigma)) + K \times \log N$$  \hspace{1cm} (7)$$

First select the number of different initial parameters, obtain the maximum likelihood estimation value of the Gaussian mixture model through the algorithm, and substitute it into the formula (7) to obtain the BIC value. Then $K$, which makes BIC value smaller, is selected as the number of Gaussian models in the Gaussian mixture model, so as to ensure the optimal number of edge gateway devices.

3.3. Solving process
Through the model established above, the objective function formulas (2) and (5) are finally obtained. When seeking the maximum value of the probability distribution, it is necessary to take the logarithm of the objective function and rewrite the objective function expression into formula (8):

$$\max \sum_{i=1}^{N} \log \sum_{j=1}^{K} \alpha_j \times f(x_i; \mu_j, \sigma_j)$$  \hspace{1cm} (8)$$

The parameters involved in the model need to be estimated by using the input sample nodes as variables. This article will adopt the Expectation-Maximization (EM) iterative algorithm[4]. Through this algorithm, the parameters in the Gaussian mixture model are expressed by an iterative formula.

The solution process of the EM algorithm[4] is divided into two steps.

The first step (Step E): First, calculate the posterior probability based on the parameter values or initial values iterated in the previous step. The iterative expression is (9):

$$\phi_{ij}^{(t+1)} = \frac{\alpha_j \times f(x_i; \mu_j, \sigma_j)}{\sum_{j=1}^{K} \alpha_j \times f(x_i; \mu_j, \sigma_j)}$$  \hspace{1cm} (9)$$

The second step (Step M): Calculate the parameter values of the Gaussian model through the values of the first step. Repeat these two steps until the absolute value of the subtraction of the two parameter values is less than the given constant value. It is considered that the model iteration is completed and the objective function has reached the extreme value.

Record the parameters of each Gaussian model as $\phi = \{\phi_1, \phi_2, \cdots \phi_k\}$, where each model is $\phi_j = (\alpha_j, \mu_j, \sigma_j)$. After the initial value is given, first calculate $\phi_0$, then calculate $\phi_1$ through the
sample node data \( X \), repeat the above steps, in this way, the parameters of each Gaussian model component can be expressed as: 
\[
\phi_j^{t+1} = (\alpha_j^{t+1}, \mu_j^{t+1}, \sigma_j^{t+1})
\]

Iterate according to the above iterative formula, and ensure that the difference between the two iterations is less than any constant or specified value, the iteration ends. After iteratively calculating the parameter value of each Gaussian component, calculate the probability of the sample node on each Gaussian component, select the number of Gaussian components by Bayesian information criterion[7], and finally compare all the sample nodes with the objective function of load balancing. For function values under different clustering effects, select the smaller function value as the final clustering basis.

4. Example verification and analysis
This part is mainly to experiment on the deployment method of edge computing gateway equipment based on Gaussian mixture model proposed in this article. While giving the experimental results and analyzing these results, it proves the advantages of this deployment method.

First, 500 sample nodes are randomly generated through the python language, and after these sample nodes are brought into the model, iterate with 2, 3, 4, 5, and 6 Gaussian components respectively.

After each parameter value is iterated through the EM algorithm [4], the probability value is obtained in turn according to the algorithm steps, and then the minimum value of the load balancing objective function is obtained through comparison. This distribution is the basis for deployment we need.

According to the randomly generated sample nodes, the experimental results show which clustering effect is used to deploy our edge computing gateway equipment.

Bayesian information criterion index is added into the code to collect the BIC value and the number of Gaussian components of each mixed model, and the final selected Gaussian mixture model can be determined by comparing the BIC value. It can be seen from Figure 2 that when \( k=4 \), the BIC value of the sample node is the smallest, indicating that the Gaussian mixture model has the best effect, that is, 4 Gaussian components are selected to be deployed in the sample model as a gateway device to provide edge computing services.

![Figure 2. Bias information criterion numerical diagram.](image-url)

To ensure load balance, it is necessary to ensure that the number of sample nodes contained in each Gaussian component does not differ too much. Consider selecting 4 Gaussian components to cluster based on the BIC value, and calculate the clustering result that minimizes the objective function of load balancing, so as to achieve the purpose of load balancing.
In the case that the number of samples is constant, the fitting effect of the Gaussian mixture model will be different with the different distribution of the sample nodes, so the number k of Gaussian components corresponding to the minimum BIC value will also change. Figure 3 shows that in different distributions composed of 500 sample nodes, 2-20 Gaussian components are used to fit the BIC values recorded by the distribution. It can be seen from the figure that distribution 1 has the best fitting effect when k=4, while for distribution 2, when k=6, the BIC value is the smallest and the fitting effect is the best.

The above experimental data shows that different distributions have different fitting effects. Under no specific conditions, the method of using the BIC value as an indicator to determine the number of Gaussian components to fit the distribution has a good effect. After the above analysis, it can be seen that by adding the BIC value as an indicator to the algorithm, the optimal choice can be made for the number of Gaussian components. The number of Gaussian components corresponds to the number of edge computing gateway devices deployed, indicating that compared with other clustering algorithms, the initial number of clusters needs to be set based on empirical values, the edge computing gateway deployment method based on the Gaussian mixture model used in this paper can weaken the effect of the initial value, and the number of edge gateway devices can be optimized without the user having to set the initial value multiple times.

5. Conclusions
In this paper, an edge computing system scheme based on microservice architecture is proposed, and an edge computing gateway device deployment method based on Gaussian mixture model is mainly introduced for this system. This method can more accurately represent the relationship between the mobile terminal and the clustering center by calculating the probability of each sample node on each Gaussian model component. In practice, we can fit any distribution by changing the number of model components, and we can also use any reasonable mixture distribution to define the data model. Therefore, the Gaussian mixture model is more practical and more convenient than other algorithms. The gateway device deployment method described in this article can realize the clustering of sample nodes and the selection of clustering centers, and can realize the reasonable deployment of service nodes. The verification of an example proves the rationality of the gateway deployment method proposed in this article.
Acknowledgment
This work is supported by the science and technology project of STATE GRID CORPORATION OF CHINA-Research and Application of Oriented Cluster Autonomous Method for Cross-professional Perception Data (SGLNFS00HLJS2000620).

References
[1] Varghese B, Wang N, Barbhuiya S, et al. 2016 Challenges and Opportunities in Edge Computing[J]// IEEE International Conference on Smart Cloud 20-26
[2] Ma Jijun, Jia Xueqin, Shou Yanbo, etc. 2018 Industrial data collection based on edge computing [J] Information Technology and Network Security 37(4) 91-93
[3] Ren Bingyu 2020 Real-time data management gateway based on edge computing technology[J] Electronic world 16 57-58
[4] Zhao L, Liu J, Shi Y, et al. 2017 Optimal Placement of Virtual Machines in Mobile Edge Computing[C]// GLOBECOM 2017- 2017 IEEE Global Communications Conference. IEEE 1-6
[5] Wang W, Zhao Y, Tornatore M, et al. 2017 Virtual machine placement and workload assignment for mobile edge computing[C]/ IEEE, International Conference on Cloud NETWORKING.IEEE
[6] Lu Xiaodi 2019 Design and implementation of mobile edge computing platform [D] Xi an: Xidian University
[7] Deng M, Tian H, Lyu X 2016 Adaptive sequential offloading game for multi-cell Mobile Edge Computing[C]// International Conference on Telecommunications. IEEE 1-5
[8] Yu Liangliang, Xu Xin, Chao Jun, etc. 2020 Research on Edge Computing System Based on Microservice Architecture[J] Industrial control computer 9 120-121
[9] Wu Hongfei 2020 Research on Microservice Scheduling Algorithm Based on Edge Computing[D] Chengdu:University of Electronic Science and Technology of China
[10] Zhao Ji 2019 Research on the construction and deployment of edge computing gateway equipment[D] Da lian: Dalian University of Technology