Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 6, pp. 5372–5382, DOI 10.12694/scpe.v25i6.3381

AN INTERNET OF THINGS TASK SCHEDULING FRAMEWORK BASED ON AGILE VIRTUAL NETWORK ON DEMAND SERVICE MODEL

QIQUN LIU*

Abstract. In order to improve the efficiency of cloud computing resource utilization and avoid the problem of computing resource allocation and scheduling lagging behind load changes, the author proposes a cloud computing resource on-demand allocation and elastic scheduling method based on network load prediction. Firstly, the author takes the network load data of Wikimedia as the research object and proposes an adaptive two-stage multi network model load prediction method based on LSTM (i.e. ATSMNN-LSTM load prediction method). This method can classify the network load data into climbing and descending types based on the trend and characteristics of the input network load data, And adaptively schedule the input network load data to the LSTM load prediction model that matches its type for prediction based on the classification results. The author proposes a maximum cloud service revenue computing resource quantity search algorithm based on network load prediction (i.e. MaxCSPR-NWP algorithm), which aims to improve cloud service revenue as the optimization objective. Under the premise of ensuring task service quality and system stability, the algorithm allocates cloud computing resources on demand and flexibly schedules them in advance based on the predicted network load results. The experimental results show that the ATSMNNLSTM load prediction method proposed by the author can obtain more accurate network load prediction results compared to other load prediction methods, and the MaxCSPR-NWP algorithm, which is based on network load prediction and is capable of effectively converting the network load prediction results into the required number of cloud servers, is the maximum cloud service revenue computing resource quantity search algorithm proposed by the author, not only does it achieve the early allocation and scheduling of cloud computing resources, thereby avoiding the impact of lagging behind in computing resource allocation and scheduling due to load changes on the quality of cloud computing task services and resource utilization efficiency, at the same time, it has also achieved on-demand allocation and flexible scheduling of cloud computing resources with the goal of improving cloud service revenue.

Key words: Task scheduling, Calculate resource allocation, Load prediction, Network on-demand services

1. Introduction.

Cloud computing, as a new computing model, aims to change the occupancy and usage of traditional computing systems. Cloud computing organizes and aggregates computing and communication resources in a networked manner, providing users with computing resources that can be reduced or expanded in scale through virtualization, increasing the flexibility of users in planning, purchasing, owning, and using computing systems [1]. In cloud computing, the core issue that users are concerned about is no longer the computing resources themselves, but the services they can obtain. From this perspective, it can be considered that service issues (provision and use of services) are the core and key issues in cloud computing. Cloud computing provides services to a large number of users through a unified interface by managing, scheduling, and integrating various resources distributed on the network. For example, with the help of cloud computing, user applications can process terabytes or even petabytes of information content in a very short period of time, achieving the same powerful performance as supercomputers. Users use these services on an on-demand basis, realizing their dream of providing computing, storage, software, and other resources as a common facility. Cloud computing includes two sets of concepts: cloud computing tools (hardware, platform, software, and so on). And the data service design of the system - cloud application. Implementing cloud computing services is essential. In addition to Amazon's infrastructure services, Google's application engine services, Microsoft's Azure service platform, etc., have distributed data storage and processing systems, such as open-Hadoop, also provide horizontal workflow services for storing and processing massive data.Meanwhile, more and more application developers can start developing and deploying various services and applications on cloud computing platforms. It can be predicted that there will be more and more service resources available on the Internet, so how to implement on-demand personalized services in cloud computing is of great significance. The service

* School of Tourism Management, Henan Vocational College of Agriculture, Zhengzhou, Henan, 451450, China (19921302270@hnca.edu.cn)

issues in cloud computing involve not only the requirements that users expect to achieve, but also the functions and performance that cloud computing service providers can provide.

Cloud computing converts traditional services into ”charging when you go” service models, allowing for computing resources such as water, electricity, and natural gas. On demand compensation with day-to-day usage greatly reduces the deployment and performance of network services. At the same time, due to restrictions in computing resources and power consumption, terminal resources can not utilize complex services. At the same time, terminal resources can not consume complex services. The terminal equipment can transfer tasks to cloud computing center (hereinafter called cloud center) by the network to get enough resources and make a good sense of (QE) of tasks, such as computing accuracy, scalability, and so on.Therefore, cloud computing not only provides a new direction for traditional Internet services, but also provides an unprecedented opportunity for the development of Internet applications, making Internet services such as image recognition, audio processing, virtual reality widely used. Furthermore, cloud computing has also promoted the rapid development of Internet of Things (IoT) and mobile Internet, replacing the traditional Internet of Things as the new CoT model of Internet of Things. However, in practical situations, due to the distance of the cloud center from terminal devices and users, unloading tasks from terminal devices and users to the cloud center can cause significant network latency, making it difficult to ensure the quality of service (QoS) of tasks. In addition, due to the large-scale clustered system architecture of cloud centers and the adoption of centralized management mode, massive task offloading will also increase the cost and complexity of cloud computing resource management. In order to address the aforementioned issues, a new computing service model - fog computing - has been proposed. Fog computing is composed of a large number of computing devices deployed on edge networks, which can collaborate with each other to enable tasks to be processed in edge devices (i.e. fog nodes) close to the terminal. This not only reduces network latency and cloud center load for tasks, but also enhances the mobility and security of network services. Therefore, fog computing can be considered as an extension of cloud computing, serving as an intermediate layer between terminal devices and cloud centers to deploy lightweight computing devices closer to terminal devices and users, providing high elasticity and fast response computing services to meet the real-time requirements of latency sensitive tasks.

2. Methods.

2.1. Application Architecture of On Demand Services in Cloud Computing.

The application architecture of on-demand services based on cloud computing is shown in Figure 2.1, which specifically includes:

(1) Cloud infrastructure layer. The model supports multiple cloud centers, not only from internally controllable clouds, but also from external third-party cloud resources that meet corresponding service level agreements. The cloud platform will consolidate these capabilities to provide a unified cloud service for the above modeling and implementation [2]. The Cloud hardware supports a wide range of devices, including hardware, network devices, processors and non-procedural data storage, as well as other basic software components.

(2) Service resource management and monitoring layer. The various resources provided by the cloud infrastructure layer, as well as service resources for specific business applications, including cloud service resources from third parties, can be uniformly registered in service resource management and provided to the public in the form of services [3,4]. This layer mainly solves the effective monitoring and management of the resources management in large distribution areas, provides support for the resources management and the states for the above resources and the services needed. According to different types of resource management and monitoring requirements, the resource structure, management behavior, and monitoring strategies can be defined to achieve resource management and monitoring.

(3) Programming Framework and Running Engine Layer for Cloud Services. This layer provides the cloud service runtime support for the top of the base platform and the service runtime, as well as engine support for the cloud service runtime. The continued application of cloud computing in many enterprise environments will lead to the rapid development of cloud based application programming interfaces (APIs) and services. A specific application typically involves multiple services, requiring a runtime support engine that integrates consolidated databases and various types of services, including pre-defined service models, application programming interfaces, automatic application services for large and dynamic environments, and efficient and reliable application programming in high-level situations.

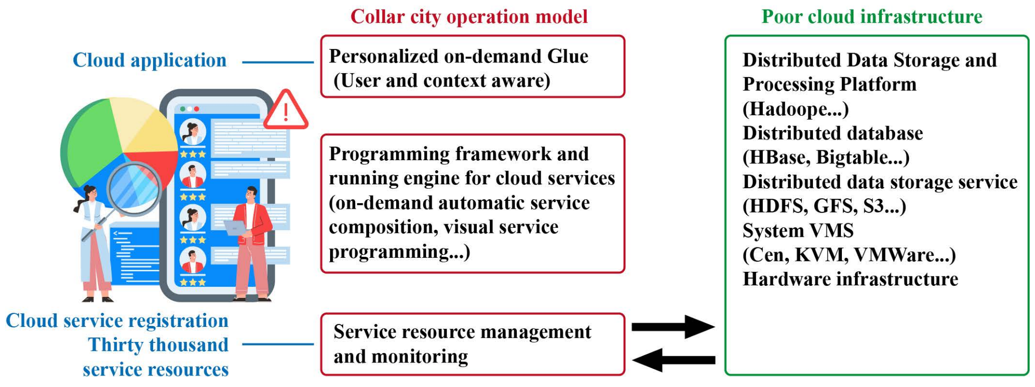
Collar city operation model
Poor cloud infrastructure
Cloud application
Personalized on-demand Glue (User and context aware)
Programming framework and running engine for cloud services (on-demand automatic service composition, visual service programming...)
Service resource management and monitoring
Cloud service registration Thirty thousand service resources
Distributed Data Storage and Processing Platform (Hadoope...)
Distributed database (HBase, Bigtable...)
Distributed data storage service (HDFS, GFS, S3...)
System VMS (Cen, KVM, VMWare...)
Hardware infrastructure

Fig. 2.1: An on-demand service application architecture diagram based on cloud computing

(4) Personalized on-demand service layer. The problem to be solved at this stage is how to provide users with personalized cloud services that are "real-time, on demand" and based on different cloud services. It mainly deals with two aspects: how to encourage users to accurately and easily describe their needs, thus realizing the discovery, matching, and approval of services according to users' needs. How to visualize the state of available resources, user context, and other information in cloud computing, in order to provide a cloud service that can be adapted to change the state of this information.

2.2. Cloud computing resource on-demand allocation and elastic scheduling system architecture based on network load prediction. The on-demand allocation and elastic scheduling system of cloud computing resources based on network load prediction mainly includes the following parts:

Cloud computing resource pool: This computing resource pool is composed of cloud servers (i.e. virtual machines, Virtual Machines (VMs)) running within a physical server cluster. The computing resources of cloud servers (such as CPU, memory, storage, and bandwidth) can be configured on demand through virtualization technology according to needs. This computing resource pool is generally considered as the Infrastructure as a Service Layer (IaaS Layer) of cloud computing, providing support for computing resources for cloud services. Task scheduling+network access service component: This component provides load balancing services for network tasks. Tasks that arrive at the cloud center are scheduled to the cloud server in the cloud computing resource pool according to certain rules (such as polling). Cloud Center Management Platform: This platform is used to manage the computing resource pool of the cloud center. The resource monitoring service is used to monitor and collect operational information of cloud computing resource pools, such as computing resource load information, physical server health status, and hardware configuration information, and store the collected data information in the platform's database. These data will also be used to assist computing resource scheduling services in finding suitable target physical servers to start and shut down cloud servers. The access control service provides network interfaces and identity authentication for cloud service administrators, and is responsible for receiving cloud server startup or shutdown instructions issued by administrators. After receiving resource management instructions, the cloud center resource management service will first find the target physical server based on specified policies (such as minimum number of physical computing nodes, balancing the number of cloud servers, and random physical server selection policies). Then, the computing resource scheduling component will send the management instructions for computing resources to the target physical server to start or shut down the cloud server. Compared with general cloud center management platforms, in order to achieve on-demand allocation and elastic scheduling of cloud computing resources based on network load prediction, the following components need to be extended: load monitoring component: This component can periodically obtain statistical data information on the number of network task requests and load (i.e. network load) through the load balancing service interface, providing necessary historical data information for network load prediction. Load forecasting service components: This component predicts and outputs the network load value for the next time based on historical information of network load. The resource allocation service component calculates the required number of cloud computing resources (i.e. the number of cloud servers) based on optimization objectives (such as service quality assurance or improving service revenue). This service component starts or shuts down the required number of cloud servers by calling the computing resource scheduling service in the cloud center. Based on the structure of the above system, the author mainly conducts specific research on the core technologies of load forecasting service components and computing resource allocation service components - load forecasting methods and computing resource quantity search algorithms.

2.3. Design of on-demand allocation and elastic scheduling methods for cloud computing resources based on network load prediction.

(1) Mathematical modeling of task processing in cloud servers. The cloud server runs in the form of a virtual machine within the physical server of the cloud center resource pool, and the virtualization software of the physical server abstracts the CPU memory, storage, and network cards of the physical layer into the computing resources of the virtual machine, such as vCPU, vMemory, vDisk, and vNIC (Virtual Network Card). When a task is offloaded to the cloud center, the task scheduling service in the cloud center will allocate the task to the target cloud server for processing; After the task arrives at the cloud server, it will be queued and buffered within the cloud server. Then, the computing unit composed of vCPUs processes the task in the order in which it is queued; When a task cannot meet its deadline requirements, it will be removed from the task queue of the cloud server and discarded, resulting in task loss [5,6].

Due to the large-scale clustered system architecture and centralized management mode of cloud centers, the process of cloud computing resource allocation and scheduling has a high complexity. This not only results in high latency in the process, making it difficult to achieve real-time performance similar to that of fog node computing resource allocation and scheduling, but also increases the cost of the process. Therefore, the cycle interval for adjusting and billing cloud computing resources is generally set to a larger time interval, with a typical cycle interval of 1 hour, which is consistent with the network load information collection cycle interval of Wikimedia services. It is not only difficult to use a real-time queuing model to mathematically model the task processing process of cloud servers, It is also difficult to use random queuing models to directly mathematically model the processing process of cloud server tasks within one hour. The network load data of Wikimedia services is a large-scale web service request task, and does not follow a time homogeneous Poisson process in hours (i.e. follows a non time homogeneous Poisson process). However, the statistical distribution of network load within one hour can be obtained by using load monitoring services to analyze past network load data. The Wikimedia network load calculated by the load monitoring service follows a Homogeneous Poson Process with consistent mean for each small granularity unit time within an hour. In addition, in order to maintain the completeness of theoretical analysis, the author assumes that the length of task execution time follows an exponential distribution: and in practical situations, this statistical distribution can be obtained by analyzing historical task execution time data. Therefore, the author describes the task processing process of the cloud server within one hour of each Δ_τ as an MM/1/K random queuing model. The MM/1IK random queuing model uses a First Come First Serve strategy to queue tasks, where 1 represents a processing unit composed of cloud server vCPUs; K is the capacity of the task queue. When the number of tasks exceeds the queue capacity, tasks cannot enter the queue and suffer losses. According to the requirements of task quality of service (QoS), the task queue capacity of each cloud server is calculated as follows:

$$K = \frac{dp^{max}}{\overline{et}} \tag{2.1}$$

Among them, dp^{max} is the allowable execution deadline of the task, which means dp^{max} is equal to the deadline of the task minus the time when the task arrives at the cloud server; $\overline{et}$ is the average execution time of the task.

According to the average execution time of tasks on the cloud server, the average number of tasks processed by the cloud server within Δ_τ is:

$$\mu = \frac{\Delta_\tau}{et} \tag{2.2}$$

The average task processing load of the cloud server within Δ_τ is calculated as follows:

$$\rho(\Delta_\tau) = \frac{\overline{\lambda(\Delta_\tau)}}{\mu} \tag{2.3}$$

Among them, $\overline{\lambda(\Delta_\tau)}$ is the task arrival rate of each cloud server within Δ_τ (i.e. the average number of tasks arriving at the cloud server) [7].

Cloud services generally consist of task scheduling services and computing resource pools. Among them, task scheduling services can be composed of load balancing components. Task scheduling services schedule tasks to the target cloud server according to corresponding strategies, using the most commonly used RoundRobin (RR) strategy as the task scheduling strategy, which schedules tasks to each cloud server in the computing resource pool in a polling manner.

Therefore, when the total number of task arrivals in the t-th cycle interval is x (t) and the number of cloud servers in the computing resource pool is Nv (t), the number of task arrivals per cloud server in the t-th cycle interval is:

$$x_v(t) = \frac{x(t)}{N_v(t)} \tag{2.4}$$

(2) Design of a maximum cloud service revenue computing resource quantity search algorithm based on network load prediction. The revenue of cloud service is the main concern of Internet business. The revenue of cloud service not only determines the sustainable development of Internet business, but also affects the development direction of Internet business. Therefore, the author aims to maximize the revenue of cloud services and conducts research on on-demand allocation and flexible scheduling of cloud computing resources based on network load prediction. Cloud computing transforms traditional computing resources into an on-demand and paid service model, avoiding human and material costs such as purchasing computing hardware, system maintenance, energy and power, and cooling. Therefore, from the perspective of cloud service providers, the calculation of cloud service revenue only needs to consider the rental cost of cloud servers, the service revenue of tasks, and the loss cost of tasks. If C1 in C=[C1, C2] represents the rental cost of each cloud server (in hourly intervals), and C2 represents the revenue of each task, then the revenue of cloud services during the t-th interval is calculated as follows:

$$profit(t) = m \cdot N_v(t) \cdot [C_2 \cdot \overline{\lambda(\Delta_\tau)} - C_2 \cdot \overline{\lambda_{loss}(\Delta_\tau)} - C_1/m] \tag{2.5}$$

Among them, $C_2 \cdot \overline{\lambda(\Delta_\tau)}$ in formula 3.5 is the average total revenue of each cloud server's tasks within Δ_τ, $C_2 \cdot \overline{\lambda_{loss}(\Delta_\tau)}$ is the average cost of task loss for each cloud server within Δ_τ, C1/m is the rental fee for each cloud server within Δ_τ. Therefore, $[C_2 \cdot \overline{\lambda(\Delta_\tau)} - C_2 \cdot \overline{\lambda_{loss}(\Delta_\tau)} - C_1/m]$ is the average revenue of each cloud server in Δ_τ's cloud services. Based on the constraints of task service quality and cloud server stability, the maximum cloud service revenue target can be expressed as:

$$\begin{gathered} Maximizeprofit(t) \\ s.t, W_s(\Delta_\tau) \leqslant dp^{max}, \rho_e(\Delta_\tau) < U^{max} \end{gathered} \tag{2.6}$$

Among them, dp^{max} is the execution deadline of the task, and U^{max} is the maximum computing load allowed by the cloud server. Note that in formula 3.5, the revenue of cloud services is related to the total rental cost of cloud servers and the loss cost of tasks, that is, it is related to the loss value $m \cdot N_v(t)[C_1/m + C_2 \cdot \overline{\lambda_{loss}(\Delta_\tau)}]$ of cloud services, and the total number of tasks reached is an objectively determined value. Therefore, the maximum cloud service revenue target in formula 3.5-3.7 can be equivalent to the minimum cloud service loss target, that is:

$$\begin{gathered} MinimizeCost(t) = m \cdot N_v(t) \cdot [C_1/m + C_2 \cdot \overline{\lambda_{loss}(\Delta_\tau)}] \\ s.t, W_s(\Delta_\tau) \leqslant dp^{max}, \rho_e(\Delta_\tau) < U^{max} \end{gathered} \tag{2.7}$$

The convex property of formula 3.7 is related to the rental cost of cloud servers and the loss cost of tasks. Assuming m=36000, the average execution time of the task is 5ms, and the task execution deadline is 100ms. According to the typical cloud server rental fee of 1.1279 yuan/hour for Amazon Web Service (AWS), consider the following three scenarios.

1. Task high return. Considering that Wikimedia services are a typical high concurrency web service (with a task count of 10 to the power of 7 per hour), assuming a profit of 0.01 yuan per task, the loss cost of the task will become the main factor affecting the loss value of cloud services, while the overall impact of cloud server rental costs is relatively small. Therefore, in order to reduce the loss value of cloud services, a sufficient number of cloud servers are needed to reduce the cost of task loss, that is, the loss value of cloud services decreases rapidly with the increase of the number of cloud servers. When there is redundancy in the number of cloud servers, the loss value of cloud services will slowly increase with the increase of the number of cloud servers. Therefore, the loss value of cloud services is a ”weak” convex function [8].

2. Low return on tasks. Assuming that the return on each task is 0.000001 yuan, the total rental cost of cloud servers will become the main factor affecting the loss value of cloud services. In this case, the minimum number of cloud servers is the best choice for cloud computing resource allocation, and the loss value of cloud services is a monotonically increasing function.

3. The task revenue falls between high and low revenue. Assuming that the revenue for each task is 0.0001 yuan, the total cloud server rental cost and task loss cost will become the main factors affecting the cloud service loss value. In this case, as the number of cloud servers increases, although the cost of task loss decreases, the total rental cost of cloud servers increases; On the contrary, when the number of cloud servers decreases, although the cost of task loss increases, the total rental cost of cloud servers decreases, indicating that the cloud service loss value in formula 3.7 is a ”strong” convex function relative to the high return situation of the task.

Based on the above analysis, the author first proposes a Maximum Cloud Service Profit Resource Search Algorithm (MaxCSPR), which searches for the required number of cloud servers with the maximum cloud service profit as the target (equivalent to the minimum cloud service loss target) based on the input network load data. In order to adapt to the three scenarios of task benefits, the MaxCSPR algorithm includes incremental search for cloud server resources and reduced search for cloud server resources. That is, when the algorithm searches incrementally based on the number of cloud servers and the loss of cloud services also increases, it indicates that the incremental search direction causes the loss value of cloud services to increase with the increase of the number of cloud servers. At this time, it should search in the direction of decreasing the number of cloud servers, perform a search to reduce the number of cloud servers, and vice versa. In order to avoid the limitation of empty computing resources and exceeding the maximum computing resource capacity in cloud services, the MaxCSPR algorithm sets the minimum number of cloud servers to 1 and the maximum number of cloud servers to N_v^{max}. In order to achieve on-demand allocation and flexible scheduling of cloud computing resources based on active methods, the author combines the proposed ATSMNN-LSTM load prediction method with the MaxCSPR algorithm. Specifically, the network load prediction result x ’(t+1) obtained by the ATSMNN-LSTM load prediction method is used as the network load input value for the t+1st cycle interval, replacing the input x (t+1) in the MaxCSPR algorithm, by predicting the network load and using the MaxCSPR algorithm, the cloud center can obtain the required number of cloud servers for the t+1st cycle interval in advance, in order to achieve early allocation and scheduling of computing resources and avoid the problem of computing resource allocation and scheduling lagging behind load changes. The author named the algorithm that combines ATSMNN-LSTM load prediction method with MaxCSPR algorithm as the maximum cloud service revenue computing resource quantity search algorithm based on network load prediction.

3. Result analysis.

3.1. Experimental Environment Setting. The author collected approximately 40000 hours of Wikimedia service network load data, which includes statistical values of the number of task requests with hourly intervals. Among them, 70% of the dataset is used as the training set, and 30% is used as the testing set. In order to verify the performance of the ATSMNNLSTM load prediction method proposed by the author, the experiment sets the network load data from the past 6 hours starting from the current moment as input, and predicts the network load value for the next hour; The number of neurons in the LSTM model is 25. Then, the author combines the proposed load prediction method (ATSMNN-LSTM) with the Auto Integrated Moving Average (ARIMA) load prediction method, Support Vector Regression on (SVR) load prediction method, Linear Regression (LR) load prediction method, and Single LSTM model load prediction method Compare the fully connected neural network models (ND load prediction method and ATSMNN load prediction method (i.e. replacing the LSTM model with the NN model in the ATSMNN-LSTM load prediction method).

Table 3.1: Comparison of performance indicators of different load prediction methods

	ATSMNN -LSTM	ATSMNN	LR	ARIMA	SVR	LSTM	NN
MAPE	0.0277	0.0345	0.0383	0.0537	0.0795	0.0326	0.0355
RMSE	3.7086	4.2958	4.8622	8.6442	7.814	3.9243	4.4513
(x105)R2	0.9523	0.9359	0.9179	0.7403	0.7878	0.9466	0.9312

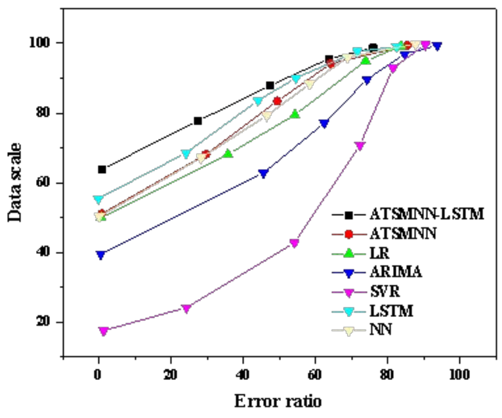

Fig. 3.1: Comparison of cumulative distribution errors of different load prediction methods

3.2. Experimental results and analysis of load prediction methods. The predicted results of the ATSMNN-LSTM load prediction method proposed by the author and other load prediction methods are consistent with the actual network load data, indicating that the ATSMNNLSTM load prediction method proposed by the author and other load prediction methods compared can effectively predict the network load change trend of Wikimedia services. In order to compare the performance of different load prediction methods and further demonstrate the performance indicators of different load prediction methods on the test dataset, as shown in Table 3.1.

As shown in Table 3.1, the MAPE and RMSE values (0.0276, 3.7085×10^5) of the ATSMNNLSTM load prediction method proposed by the author are lower than those of other load prediction methods, indicating that the ATSMNN-LSTM load prediction method can achieve lower prediction errors. The value of R2 also indicates that the R2 value of the ATSMNN-LSTM load prediction method is higher than that of other load prediction methods, and the R2 value of this load prediction method reaches 0.9523, which is closer to 1.0, indicating that the ATSMNN-LSTM load prediction method can better fit the network load change pattern of Wikimedia services. In addition, as shown in Table 3.1, the prediction accuracy of the LSTM model is higher than that of other traditional methods, and the ATSMNN-LSTM load prediction method proposed by the author further improves the prediction accuracy of network loads on the basis of a single LSTM model.

Meanwhile, the author analyzed the performance of different load prediction methods using cumulative distribution error, where cumulative distribution error is defined as follows: The input (X-axis) is the numerical proportion of the relative maximum prediction error; The output (Y-axis) is the proportion of the number of predicted data below the specified relative error ratio to the entire prediction result dataset, as shown in Figure 3.1 [9].

From Figure 3.1, it can be seen that the ATSMNN-LSTM load prediction method proposed by the author has a higher proportion of data with a relative error of less than 60% compared to other load prediction methods,

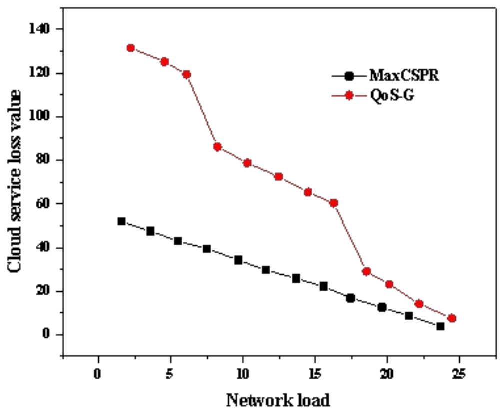

Fig. 3.2: Comparison of Cloud Service Loss Values between MaxCSPR Algorithm and QoS G Algorithm

indicating that the prediction error of the ATSMNN-LSTM load prediction method is mainly concentrated in the relative error range of less than 60% compared to other load prediction methods, that is, the proportion of data with a relative error of more than 60% in other load prediction methods is higher than that of the ATSMNNLSTM load prediction method. From Figure 3.1, it can be further observed that the difference in the proportion of relative error data between the ATSMNN-LSTM load prediction method and other load prediction methods increases with the decrease of error proportion. This indicates that the proportion of data in the ATSMNN-LSTM load prediction method with lower relative error proportion is higher than that in other load prediction methods. That is to say, the proportion of data in the low relative error range in other load prediction methods is relatively small, while the proportion of data in the high relative error range is larger, further proving that the ATSMNN-LSTM load prediction method can achieve higher load prediction accuracy.

3.3. Simulation experiment and result analysis of cloud computing resource quantity search algorithm. In order to evaluate the performance of the cloud computing resource on-demand allocation and elastic scheduling methods proposed by the author, the author further conducted simulation experiments on cloud computing resource on-demand allocation and elastic scheduling. The experimental parameter settings include: setting m=36000, the average execution time of the task is 5ms (i.e. the processing capacity of the cloud server is MIPS=1000, the average length of task instructions is 5MIMillion Instructions), and the execution period is 100ms. The upper limit of the cloud server's computing load is 90%. According to the typical price of Amazon Cloud Services (AWS), the rental fee for each cloud server is 1.1279 yuan/hour, assuming a revenue of 0.0001 yuan per task. Firstly, this section evaluates the performance of the proposed cloud computing resource on-demand allocation algorithm (i.e. MaxCSPR algorithm). Generally speaking, on-demand allocation and flexible scheduling algorithms for cloud computing resources only need to consider the guarantee of task processing performance and service quality. The author collectively refers to the cloud computing resource allocation and scheduling method with the goal of ensuring service quality as the QoS Guarantee Algorithm (QoS G). In order to evaluate the performance of MaxCSPR algorithm in cloud service loss, computational load, and task service delay, the author compared the performance of MaxCSPR algorithm and QoS G algorithm under different network loads, as shown in Figures 3.2 to 3.4.

As shown in Figure 3.2, since the QoS G algorithm aims to ensure the quality of service for tasks and does not consider the best match between the number of cloud servers and task revenue, while the MaxCSPR algorithm aims to minimize cloud service loss, the MaxCSPR algorithm can achieve lower cloud service loss values, indicating that the MaxCSPR algorithm can effectively improve the revenue of cloud services.

From Figure 3.3, it can be observed that the MaxCSPR algorithm not only aims to reduce the loss value

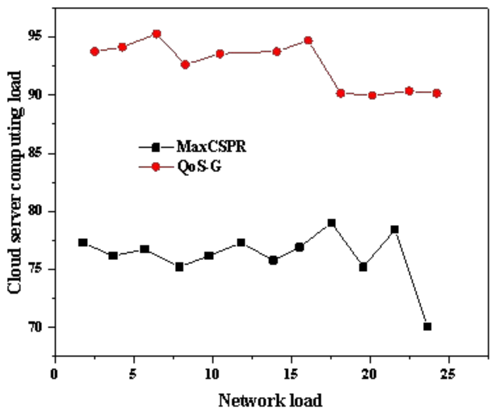

Fig. 3.3: Comparison of Cloud Server Computing Load between MaxCSPR Algorithm and QoS G Algorithm

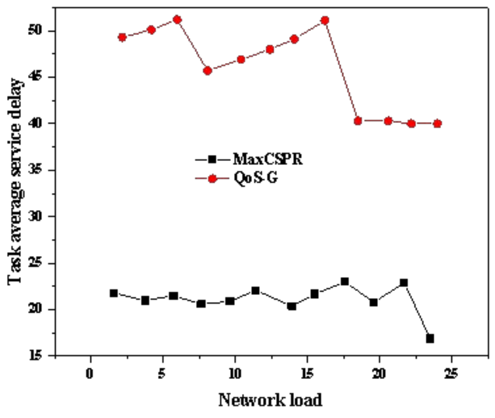

Fig. 3.4: Comparison of task average service delay between MaxCSPR algorithm and QoS G algorithm

of cloud services, but also considers the stability of the system (i.e. the upper limit of the computing load of the cloud server). Therefore, the MaxCSPR algorithm can ensure that the computing load of the cloud server is less than the required upper limit (90%), ensuring the stable operation of cloud services. Further combining with Figure 3.4, it can be seen that although the QoS G algorithm ensures the service quality of tasks (i.e., the average service delay of tasks is less than 100ms), the QoS G algorithm does not consider the stability of the system, resulting in a computing load of over 90% on cloud servers. In practical situations, this can lead to problems such as server overload or software failures. In addition, Figure 3.4 also indicates that the MaxCSPR algorithm can achieve lower task service latency. This is because the QoS G algorithm allocates cloud computing resources with the goal of ensuring the service quality of tasks. Therefore, when the cloud server meets the task execution deadline, tasks will be scheduled as much as possible to the cloud server for processing [10]; However, the MaxCSPR algorithm not only ensures the service quality of tasks, but also considers the stability constraints of the system. Therefore, the MaxCSPR algorithm achieves relatively low cloud server computing load, thereby reducing the service delay of tasks. Overall, compared to the QoS

Table 3.2: Comparison of Error in MaxCSPR Cloud Server Quantity Based on Different Load Prediction Methods and Actual Network Load

	ATSMNN -LSTM	ATSMNN	LR	ARIMA	SVR	LSTM	NN
MAPE	0.0277	0.0349	0.0382	0.0534	0.0794	0.033	0.0356
RMSE	0.7792	0.881	0.9662	1.551	1.4662	0.8214	0.8972
(x105)R2	0.9337	0.9178	0.9011	0.7449	0.772	0.9285	0.9147

G algorithm, the MaxCSPR algorithm not only ensures the service quality of tasks and system stability, but also effectively improves cloud service revenue, proving that the design of the MaxCSPR algorithm meets theoretical analysis and design requirements. Furthermore, in order to analyze the impact of network load prediction accuracy on the MaxCSPR-NWP algorithm, the author compared the error between the number of cloud servers obtained by the MaxCSPR algorithm under different load prediction methods and the number of cloud servers obtained by the MaxCSPR algorithm under actual network load data (i.e. the actual number of cloud server requirements), as shown in Table 3.2.

From Table 3.2, it can be seen that the MAPE and RMSE values of the number of cloud servers obtained by the ATSMNN-LSTM load forecasting method (0.0277, 0.7792) are lower than those obtained by other load forecasting methods, at the same time, the R2 indicator (0.9357) of the number of cloud servers obtained by the ATSMNN-LSTM load prediction method is also the closest value to 1.0 among all methods, indicating that the error between the number of cloud servers obtained by the ATSMNN-LSTM load prediction method and the number of cloud servers obtained from actual network load data is the smallest. In summary, it can be seen that in the active mode, the accuracy of load prediction is a key factor affecting the correctness of cloud server allocation. Therefore, the MaxCSPR-NWP algorithm can obtain a more accurate number of cloud servers.

4. Conclusion. In order to improve the accuracy of network load prediction, the author proposes an adaptive two-stage multi network model load prediction method based on LSTM (i.e. ATSMNN-LSTM load prediction method) to avoid the labor cost problem of manually classifying and annotating network load training data, the author achieved automatic classification and annotation of network load training datasets using first order features and K-means unsupervised machine learning algorithm. Then, based on the long-term interval characteristics of cloud computing resource adjustment and billing, the author mathematically models the cloud server task processing process within one hour using a random queuing model and traversal process. Then, a maximum cloud service revenue computing resource quantity search algorithm based on network load prediction is proposed, allowing the cloud center to obtain the required number of cloud servers in advance through the predicted load results. The experimental results show that the cloud computing resource on-demand allocation and elastic scheduling method based on network load prediction proposed by the author can not only obtain more accurate network load prediction results.

5. Acknowledgement. General Research Project of Humanities and Social Sciences in 2023 of the Education Department of Henan Province China, Study on High-quality Development of Leisure Agriculture in Henan Province Empowered by Digital Economy, No: 2023-ZDJH-135

REFERENCES

[1] Hu, Q., Wu, X., & Dong, S. (2023). A two-stage multi-objective task scheduling framework based on invasive tumor growth optimization algorithm for cloud computing. Journal of grid computing.0-(214),456

[2] Vijayalakshmi, V., & Saravanan, M. (2023). Reinforcement learning-based multi-objective energy-efficient task scheduling in fog-cloud industrial iot-based systems. Soft computing: A fusion of foundations, methodologies and applications679(23), 27.

[3] Dilek, S., Oracevic, A., Tosun, S., & Zdemir, S. (2022). Towards qos-aware resource allocation in fog computing: a theoretical model. 2022 International Symposium on Networks, Computers and Communications 678(ISNCC), 1-6.

[4] Kaur, M., Sandhu, R., & Mohana, R. (2023). A framework for scheduling iot application jobs on fog computing infrastructure based on qos parameters. International journal of pervasive computing and communications.6789(4),879

[5] Dan, F., Bo, L., & Jian, G. (2022). An on-board task scheduling method based on evolutionary optimization algorithm. Journal of Circuits, Systems and Computers.45(6783),128

[6] Akahoshi, K., & Oki, E. (2023). Service deployment model with virtual network function resizing based on per-flow priority. IEICE Transactions on Communications, 2022(435),EBP3145.

[7] Zhang, P., Wang, C., Kumar, N., & Liu, L. (2022). Space-air-ground integrated multi-domain network resource orchestration based on virtual network architecture: a drl method. IEEE transactions on intelligent transportation systems86(3), 23.

[8] Wu, G., Luo, Q., Du, X., Chen, Y., Suganthan, P. N., & Wang, X. (2022). Ensemble of metaheuristic and exact algorithm based on the divide-and-conquer framework for multisatellite observation scheduling. IEEE Transactions on Aerospace and Electronic Systems.4(23),14

[9] Fan, H., Yang, Z., Zhang, X., Wu, S., Long, J., & Liu, L. (2022). A novel multi-satellite and multi-task scheduling method based on task network graph aggregation. Expert Systems with Application.3(4),90

[10] Corallo, A., Crespino, A. M., Lazoi, M., & Lezzi, M. (2022). Model-based big data analytics-as-a-service framework in smart manufacturing: a case study. Robotics and Computer-Integrated Manufacturing, 76(5), 102331-.

Edited by: Hailong Li
Special issue on: Deep Learning in Healthcare
Received: Feb 18, 2024
Accepted: Apr 9, 2024