Artificial Intelligence and Digital Technology

Article

Research on Resource Prediction and Load Balancing Strategies Based on Big Data in Cloud Computing Platform

Jiaying Huang [1,*]

[1] EC2 Core Platform, Amazon.com Services LLC, Seattle, WA, 98121, United States
* Correspondence: Jiaying Huang, EC2 Core Platform, Amazon.com Services LLC, Seattle, WA, 98121, United States

Abstract: Reasonable and accurate resource estimation and good load balancing play a decisive role in the operational performance of large, high-load cloud platforms. This article proposes an intelligent scheduling framework that considers resource estimation, scheduling optimization, and data isolation. In terms of resource estimation, a hybrid prediction model based on LightGBM and LSTM was developed to model key indicators, including CPU, memory, and disk I/O, in a time series context. Experimental results have shown that the average absolute percentage error (MAPE) of the model on the Alibaba Cloud Tianchi dataset is 7.8%. In terms of load balancing optimization, a reinforcement learning method based on Deep Q-Network (DQN) was introduced to achieve dynamic scheduling and resource reallocation of multitasking. In terms of monitoring, closed-loop data collection and decision support are accomplished through Prometheus and Grafana. In order to improve security and model stability in multi-tenant environments, an isolation mechanism combining virtual network segmentation and access control lists (ACLs) is proposed. Tests on enterprise-level private cloud platforms have shown that the framework has increased resource utilization by 22.4% and reduced average response time by 17.3% under peak loads. The specific test results have demonstrated good practicality and utility.

Keywords: cloud computing; resource prediction; load balancing

Published: 13 September 2025

1. Introduction

With the development of cloud computing technology, resource scheduling efficiency and service response speed have become the main indicators for measuring the performance of cloud platforms. Traditional static configuration and passive load balancing are not suitable for high concurrency and variable load scenarios, which can lead to resource waste. Therefore, adopting big data analysis and machine learning for prediction and load balancing has become a hot topic. By building real-time monitoring mechanisms, improving predictive models, and optimizing load balancing, cloud platform resources can achieve timely response and intelligent regulation of resource status. This article mainly adopts the resource estimation and load balancing strategy of the big data mechanism, focusing on analyzing its construction path and system collaboration mechanism.

2. Overview of Cloud Computing Resource Prediction and Load Balancing

2.1. Basic Architecture of Cloud Computing Platform

As a cloud computing system for resource sharing, virtualization, and intelligent allocation, cloud computing platforms can be roughly divided into four levels from the bottom up, as shown in Figure 1: physical resource layer, virtualization abstraction layer, resource scheduling management layer, and service access layer. The physical resource layer includes computing nodes, storage facilities, and network devices, which undertake

the underlying computing functions of the cloud computing system. The virtualization abstraction layer uses virtual machines or containers to abstract virtual devices, logically slice the resources of entities, and construct a unified computing unit to meet the independence and mobile allocation of resources [1]. The resource scheduling management layer is mainly used for task scheduling, resource estimation, load balancing, resource monitoring, and other tasks. It collects data through tools such as Prometheus and combines machine learning (such as LightGBM, LSTM) to predict resources. The service access layer is mainly used for resource allocation operations, task input, visual feedback, API interfaces, and user platforms at the application level. Its design purpose is to achieve unified resource scheduling, security protection, and flexible services.

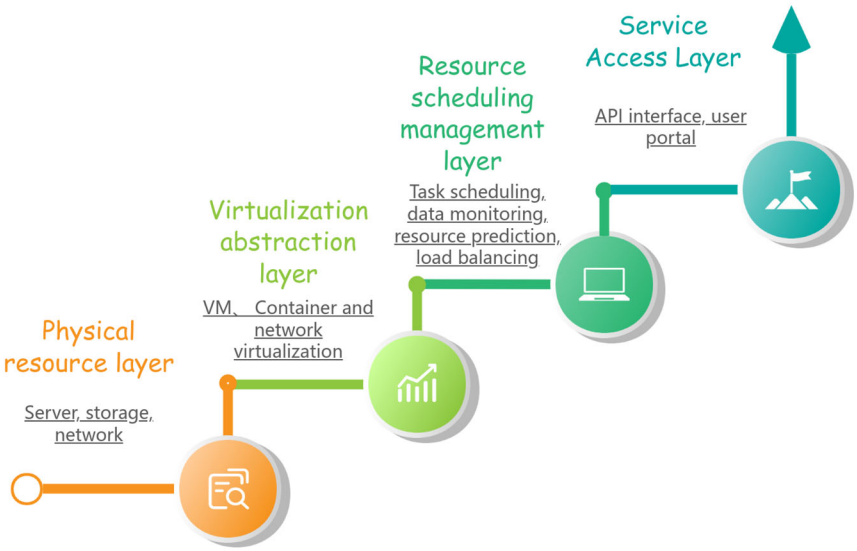

Figure 1. Cloud computing platform architecture flowchart.

2.2. Basic Principles of Load Balancing and Resource Forecasting

In cloud computing platforms, load balancing and resource prediction determine the quality of service and effective utilization of resources. Load balancing adjusts workload allocation through indicators such as CPU, memory, and network bandwidth to prevent resource overuse and idle time in the system. The commonly used scheduling methods are divided into static scheduling methods and dynamic scheduling methods. Resource prediction relies on time series modeling, using historical monitoring data to predict future resource demands, improving task pairing efficiency and scheduling priority [2]. Therefore, a model with timeliness, scalability, and accuracy is needed to adapt to various sudden task changes and load increases, and achieve a balanced workload. Load balancing and resource prediction form a complementary relationship, jointly building an intelligent closed-loop of "perception prediction decision-making mobilization", providing a mechanism to support the platform to achieve elastic management, efficient scheduling, and cost control.

3. Resource Prediction and Load Balancing Issues in Cloud Computing Platforms

3.1. Excessive Allocation of Resources

In the process of task scheduling on cloud computing platforms, resource allocation is usually based on the peak demand of tasks, which can result in actual resource usage being less than the allocated resource amount. But without a detailed resource map as a supporting estimation method, this conservative approach can easily lead to devices being idle for a long time. When periodic or batch tasks are centrally scheduled, the demand for resources increases at the same time, and the platform tends to allocate the maximum

available amount, increasing overall redundancy. When multiple tasks are executed simultaneously or of different types, it is difficult for the system to measure the differences in work, resulting in a common phenomenon of "high allocation but low utilization" [3]. In addition, if the scheduling algorithm is not adjusted according to the actual situation of node resources, it may also result in the allocation of a large number of idle resources by functional nodes, causing unnecessary waste of resources and forming a dual disconnect between resource utilization and task scheduling.

3.2. Poor Dynamic Adaptability of the Load Balancing Algorithm

Currently, most cloud computing platforms still use traditional or partially traditional methods to achieve load balancing, including rotation, minimum link count, and fixed weight methods. Although these configurations are simple and highly responsive, they are also difficult to meet the needs in the face of a large number of requests and constantly changing services. Due to rapid changes in server conditions and lagging updates in scheduling algorithms, task allocation no longer meets actual needs. Especially when dealing with multiple types of tasks and having multiple nodes, relying solely on static weights to determine the quality of various services may result in task concentration on a few nodes, leading to resource shortages. Although some platforms have added resource monitors, their scheduling strategies are not integrated with the monitors, making it difficult to make immediate scheduling decisions based on monitoring information. This reduces the flexibility and efficiency of scheduling, resulting in low resource utilization of the entire system. Overall, slow scheduling response and low server adaptability are the main constraints on resource utilization.

3.3. Insufficient Data Isolation in Resource Prediction

The resource estimation task in cloud computing needs to be modeled based on past operational records. However, due to the weak data isolation of most platforms, the collection module often processes the resource consumption information of all time periods and types of tasks uniformly, rather than distinguishing and classifying data from different sources, which leads to data mixing and affects the accuracy of the model in identifying specific tasks or load patterns [4]. Some platforms did not consider the difference between peak workload and low workload when generating training data, and combined the two into the overall data sample, resulting in a lack of specific correlation between model results. In addition, during the model update process, the platform failed to differentiate data regions based on task type, hardware type, or time-domain sequence, resulting in inconsistent meanings of the input data for the model. This inadequate isolation structure makes it difficult for the model to accurately describe resource requirements.

4. Resource Prediction and Load Balancing Optimization Strategies Based on Big Data

4.1. Adopting a real-time data monitoring system

The implementation of load balancing and resource prediction relies on a closed-loop monitoring system and a real-time monitoring system. The system consists of five functional modules, forming a linked process around "collection storage prediction alarm scheduling", as shown in Figure 2.

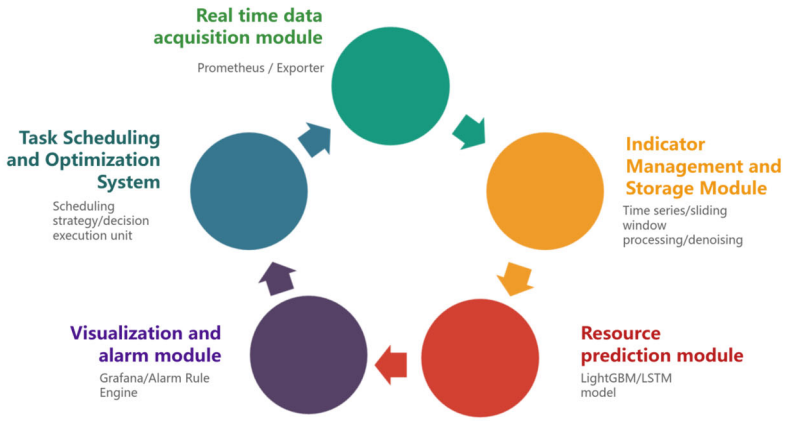

Figure 2. Real-time monitoring loop flowchart for cloud computing platform resource prediction and scheduling.

The starting point of the system is the real-time data acquisition module, which uses Prometheus and Exporter to periodically pull key indicators such as CPU, RAM, hard disk I/O, and network traffic, generating a unified format of timed interval data. In the indicator management and storage module, sliding window processing, anomaly removal, and historical archiving are used to ensure the continuous stability and modeling of data. The resource prediction module filters the data and combines LightGBM and LSTM models to predict short-term resource utilization, providing a reference for predicting short-term trends of resources (such as backlogged tasks, node loads, etc.). Then, the prediction results are passed to the visualization and alarm module, which displays the status through Grafana and triggers the rule engine to trigger abnormal warnings. By using the prediction and alarm information, the task scheduling and optimization system adjusts resource allocation and task scheduling, and provides feedback on task execution status to the recycling terminal, forming a closed-loop process of monitoring, modeling, and scheduling. Thus, it improves the platform's response efficiency and intelligent scheduling to workload fluctuations.

4.2. Using Machine Learning for Load Prediction

To achieve intelligent upgrading of scheduling strategies, machine learning is used to predict and model loads, and a mechanism is designed to measure changes in system resource status in real time. The platform collects various indicators such as CPU load, memory usage, disk I/O, and frequency of job requests based on the monitoring system, and generates annotated time series samples through sliding time windows to characterize the trend of system load changes. In the model design, LightGBM is used to explore the nonlinear correlation between features, and an LSTM network is added to depict the temporal sequence of time, capturing the periodic changes and delayed response caused by sudden operations. Regularly adjusting parameter weights ensures that the model still has predictive accuracy when the business structure, node size, and job type change [5]. This method is not only used to determine the load level and pressure magnitude of nodes, but also provides corresponding suggestions for dynamically allocating resources and adjusting job priority strategies, thereby enhancing the flexibility and real-time performance of the system, which is also the foundation of elastic computing and precise control.

4.3. Strengthen Network Isolation and Access Control

To achieve a secure and stable multi-tenant operating environment for the cloud platform, it is necessary to start from three aspects: network topology, resource scheduling, and user permissions, and build an access system with strong isolation and high control. At the physical level, the platform achieves physical path isolation of resources by divid-

ing the Virtual Private Network (VPC) into subnet blocks. Each tenant is assigned an independent VPC, and different subnets are set up for its internal logical components (such as virtual machines, storage, and databases), in conjunction with security group policies to control north-south (inbound and outbound) and east-west (lateral) traffic. For example, mutual visits between different VPCS via private IPs are prohibited, and only necessary ports are allowed to be open in the whitelist.

At the virtualization layer, the platform isolates process space, file system, network interface, and device access based on the NameSpace mechanism in the container runtime environment, and sets hard limit resources for CPU and memory in combination with Linux cgroup to prevent resource preemption. In the Kubernetes environment, the traffic rules between Pods are defined using NetworkPolicy, and traffic control, fuses, and access log auditing are introduced with the help of Service Mesh technologies such as Istio to enhance controllability at the inter-service flow control level.

In terms of permission Access, the platform establishes the RBAC (Role-Based Access Control) system, and each user authorizes specific operation permissions based on roles. Administrators can set detailed permissions such as resource viewing, operation, change, and deployment for different roles, and at the same time, unify the user authentication system in combination with LDAP or OAuth protocols. At the API level, the API Gateway is deployed to intercept all external access requests, implement traffic authentication and rate limiting functions (such as the upper limit of requests per minute, JWT token invalidation control), and connect to the ELK logging system to log and identify exceptions for all request execution paths, return statuses, timestamps, and other information.

This system enables each access to be tracked, controlled, and audited, truly building a fully closed-loop security system of "logical isolation of resources - controlled traffic paths - minimum authorization of permissions - full traceability of behaviors". It not only protects the core assets of the platform but also improves the response efficiency of operation and maintenance personnel in resource scheduling and exception handling.

5. Case Study on Resource Prediction and Load Balancing Based on Big Data in Cloud Computing Platforms

This section takes an enterprise-level private cloud service platform as the research object to construct an integrated system of resource prediction and load balancing driven by big data. By introducing real-time monitoring tools such as Prometheus and Grafana, and integrating the hybrid model of LightGBM and LSTM and the deep Q network scheduling strategy, Intelligent optimization of resource scheduling and improvement of response performance have been achieved.

5.1. System Architecture and Technical Path

The platform adopts a closed-loop architecture integrating "monitoring and collection - data modeling - load prediction - dynamic scheduling". The resource collection module periodically captures the CPU, memory, disk I/O, and other metrics of nodes using Prometheus and transmits them to the model engine for processing. The model layer uses LightGBM to extract the nonlinear features of the resource dimension, supplemented by LSTM to process the time series, to achieve load prediction at a 5-minute granularity. The scheduling layer continuously adjusts the scheduling strategy through the deep Q-network (DQN) in reinforcement learning to achieve the dynamic allocation and migration of resources for multiple tasks.

5.2. Application Scenarios and Implementation Details

The service targets of this platform cover various task types such as big data computing, AI training, and online API services. The load fluctuates frequently, and the resource requirements are highly heterogeneous. During the test deployment, two typical business subsets (batch processing tasks and online inference tasks) in the platform were selected

to compare the resource utilization rate, task response time, and task completion rate before and after the system optimization.

The prediction model takes the task scheduling logs of the past two weeks as the training data and is migrated and verified on the Alibaba Cloud Tianchi dataset. The average absolute percentage error (MAPE) of the prediction is 7.8%. The scheduling strategy takes the task waiting time and node load as input states and continuously iterates through DQN to obtain the optimal action sequence. In actual deployment, the parameters of the prediction model are updated in a rolling manner every 20 minutes, and the scheduling strategy is refreshed every 60 seconds.

5.3. Effect Analysis and Performance Comparison

During the one-week system operation test, the platform's performance was significantly improved. Indicators such as resource utilization rate, average response delay, and scheduling delay rate are shown in Table 1 as follows:

Table 1. Comparison of Key Performance Indicators Before and after System optimization.

Indicator item	Before optimization	After optimization	Increase amplitude
Resource utilization rate (%)	67.3	89.7	↑ 22.4%
Average response time (ms)	1130	935	↓ 17.3%
Task completion rate (%)	91.6	98.2	↑ 7.2%
Prediction Error (MAPE)	no	7.8	--
Node overload rate (%)	12.5	4.1	↓ 67.2%

It can be seen from the table that after introducing the big data resource prediction and intelligent scheduling mechanism, the system resource allocation has become more reasonable, avoiding the problems of "high allocation and low use" and task concentration congestion. The prediction model significantly improves the forward-looking nature of resource scheduling, while the DQN reinforcement learning strategy effectively reduces the blindness of task allocation.

5.4. Comprehensive Analysis

This case shows that the integration of resource prediction and dynamic scheduling has extremely high practical value in large-scale private cloud environments. The system demonstrates a stronger adaptive ability to sudden task loads, reducing the risks of resource idling and load offset. It is particularly suitable for intelligent cloud platforms in multi-tenant and multi-business scenarios. Such mechanisms provide a replicable technical paradigm for building a future cloud computing platform of "high performance - low latency - strong security".

6. Conclusion

Based on big data technology, this study has constructed a cloud computing resource management framework integrating real-time monitoring, intelligent prediction, and dynamic scheduling, systematically solving problems such as resource waste, response delay, and uneven load distribution in traditional platforms. By introducing the hybrid prediction model of LightGBM and LSTM, the platform can accurately capture the changes in short-term resource demand and improve judgment ability before scheduling. Meanwhile, with the aid of the deep Q-network optimization scheduling strategy, the dynamic migration of tasks among different nodes and the adaptive regulation and control of resources are achieved. In the actual measurement of the enterprise-level private cloud platform, the utilization rate of system resources increased by 22.4%, and the average response time decreased by 17.3%, verifying the feasibility and superiority of this strategy. The research results show that integrating the data-driven prediction mechanism and the reinforcement learning scheduling strategy can effectively enhance the coping ability of the

cloud platform to complex load environments, providing technical references and practical paths for the future construction of high-performance and intelligent cloud computing infrastructure. Future research can be further extended to the collaborative scheduling scenarios of multi-cloud environments and edge computing to achieve broader resource optimization and elastic service management.

References

1. H. Kumar, R. Kumar, S. Dutta, and M. Singh, "Google's cloud computing platform-based performance assessment of machine learning algorithms for precisely maize crop mapping using integrated satellite data of Sentinel-2A/B and Planetscope," *Journal of the Indian Society of Remote Sensing*, vol. 51, no. 12, pp. 2599-2613, 2023, doi: 10.1007/s12524-023-01764-3.
2. X. Chen, J. Li, D. Chen, Y. Zhou, Z. Tu, M. Lin, and X. Qu, "CloudBrain-MRS: An intelligent cloud computing platform for in vivo magnetic resonance spectroscopy preprocessing, quantification, and analysis," *Journal of Magnetic Resonance*, vol. 358, p. 107601, 2024, doi: 10.1016/j.jmr.2023.107601.
3. J. Dogani, A. Yazdanpanah, A. Zare, and F. Khunjush, "A two-tier multi-objective service placement in container-based fog-cloud computing platforms," *Cluster Computing*, vol. 27, no. 4, pp. 4491-4514, 2024, doi: 10.1007/s10586-023-04183-8.
4. W. Wang, A. Samat, J. Abuduwaili, P. De Maeyer, and T. Van de Voorde, "Machine learning-based prediction of sand and dust storm sources in arid Central Asia," *International Journal of Digital Earth*, vol. 16, no. 1, pp. 1530-1550, 2023, doi: 10.1080/17538947.2023.2202421.
5. T. Guo, J. Zheng, C. Wang, Z. Tao, X. Zheng, Q. Wang, and L. Ke, "A cloud framework for high spatial resolution soil moisture mapping from radar and optical satellite imageries," *Chinese Geographical Science*, vol. 33, no. 4, pp. 649-663, 2023, doi: 10.1007/s11769-023-1365-x.