European Journal of AI,
Computing & Informatics

Vol. 1 No.1 2025

PINNACLE ACADEMIC PRESS
JOURNALS, BOOKS & PROCEEDINGS

Article Open Access

Research on Improving Efficiency of Cloud Service Resource Allocation Based on Data Engineering

Fangyuan Li [1,*]

[1] Amazon Web Services, Inc., AWS Global Sales, Seattle, Washington, 98121, USA
[*] Correspondence: Fangyuan Li, Amazon Web Services, Inc., AWS Global Sales, Seattle, Washington, 98121, USA

Received: 07 April 2025
Revised: 11 April 2025
Accepted: 26 April 2025
Published: 28 April 2025

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Abstract: With the increasingly complex development of cloud computing, the ability to reasonably allocate resources will play a key role in improving the overall system performance. From the perspective of data engineering, this study analyzes the issues of collection, modeling, and control of scheduling data. It explores the technical constraints of current cloud platforms, focusing on fast response and accurate matching. The study also combines artificial intelligence models for optimizing computer structure, aiming to create a resource allocation strategy focused on high-frequency scheduling, intelligent budgeting, and dynamic collaboration. Improve the agility, controllability and intelligence of the whole resource allocation system.

Keywords: data engineering; cloud services; resource allocation; scheduling efficiency; intelligent scheduling

1. Introduction

With the rapid development of cloud computing, increasingly complex business needs put forward higher requirements for system performance and service capabilities. At the same time, the efficiency of resource scheduling has become a key issue. The traditional scheduling method has some disadvantages, such as slow response to multiple data sources and dynamic workloads, and insufficient accuracy in matching. By combining artificial intelligence-related technologies with pre-modeling, intelligent monitoring, and automatic adjustment schemes, new ideas for resource allocation are provided. Based on the data engineering model, combined with the design ideas of artificial intelligence and computer technology, this paper carried out a comprehensive study, proposed ways to improve the resource scheduling performance of cloud services, and provided a theoretical basis and scientific and technological guarantee for the intelligent resource management of cloud services.

2. Core Technical Elements of Data Engineering

2.1. Data Collection and Flow Mechanism: Build Efficient Scheduling Channels

The data collection and flow mechanisms are crucial for forming the foundation of intelligent cloud service resource scheduling. Massive cloud application scenarios have diverse and dynamically changing resource conditions, user behavior and job load information. Through intelligent data analysis and data exchange of data information, message queuing and flow processing (such as ApacheKafka and ApacheFlink) technology, Complete the fast and low latency transmission of data from the acquisition source to the scheduling engine, which has the spot-sensing load control function for data collection, and detects the dynamic characteristics in the large cluster resource environment. At the data transmission level, the distributed communication architecture ensures stable operation and scalability, maintaining the ability to transmit information reliably under high traffic or abnormal conditions [1]. Figure 1 illustrates how Flink drives data collection and process processing:

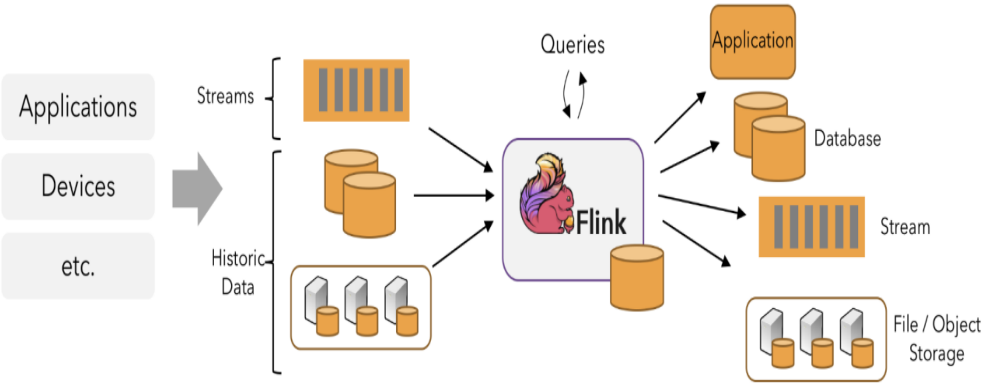

Figure 1. Flink-Driven Data Acquisition and Flow Processing Flow Chart.

2.2. Data Modeling and Storage Structure: Enhance System Computing Support

Proper metadata construction and a good data storage system are the basis for the application of resource allocation strategies. In addition to the information description of resource status, job attributes and scheduling history, it is necessary to meet the data feature extraction of intelligent algorithms in index, retrieval and prediction, and it is also necessary to meet the technologies such as dimension-based construction, graph model construction and keyword matching [2]. Based on this, high-quality data storage methods such as HBase, ClickHouse or AmazonS3 are selected according to the architecture of the computer system, which can achieve fast read and write and low latency in large-scale high-concurrency database systems. In the face of high performance and scalability of big data cluster, cold data cache and distributed index technology are used to reconstruct data arrangement in real time by machine learning, which can effectively improve the access efficiency and fault tolerance of the system.

2.3. Data Governance and Real-Time Monitoring: Ensure Stable Operation of Scheduling

The data governance and real-time monitoring mechanism ensures the stable and secure operation of the cloud service scheduling system. Among them, data governance includes data cleaning, data format standard unification, data consistency inspection, data quality control, etc. Artificial intelligence technology will be used to speed up anomaly detection and processing, ensuring data reliability and availability while preventing abnormal information from affecting scheduling decisions. Use automated governance tools such as GreatExpectations or Deequ to monitor data for real-time, accuracy, and completeness. Use machine learning methods to improve the accuracy of anomaly detection and data quality assessment to reduce system judgment errors and system configuration errors. The real-time monitoring system is based on a distributed computing architecture with intelligent components for tracking key scheduling metrics (e.g., response time, resource utilization efficiency, task completion rate). It also provides real-time monitoring and visibility alarms using tools like Prometheus and Grafana [3]. Figure 2 below is a visual monitoring image of the running status in the distributed resource scheduling working environment:

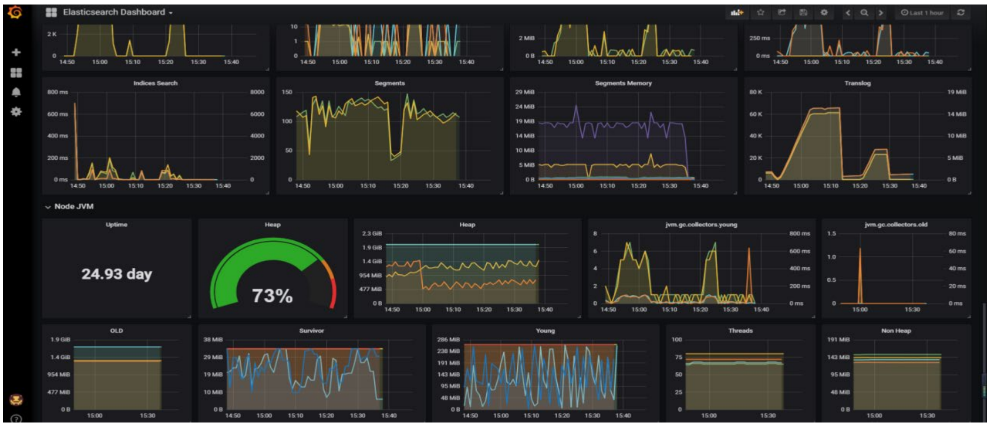

Figure 2. Visual Monitoring Diagram of Running Status in Distributed Resource Scheduling Environment.

3. Constraints on Efficiency of Cloud Service Resource Allocation from the Perspective of Data Engineering

3.1. Data Heterogeneity and Transmission Delay: Inhibit Scheduling Response Efficiency

In the cloud computing environment, the collected information comes from various sources, including user demand data, device status information, and network parameter data. The data structure is complex, including multiple forms of structure, multiple protocols and date labels, which has a direct impact on comprehensive analysis. Since the format and content of the information from all parties need to be adjusted, the data processing time is extended, leading to delayed delivery. Moreover, the cloud platform has high real-time requirements, but the time intervals for data synchronization, updates, and connection reliability vary, preventing the system from immediately obtaining accurate status information [4]. Especially in the case of cross-platform design, the problems of inter-network delay and data mismatch are more prominent, which is not conducive to making correct scheduling decisions. Table 1 below shows the main problems with data heterogeneity and transmission latency:

Table 1. Key Problems of Data Heterogeneity and Transmission Delay.

Problem type	Concrete performance
Data source diversity	Resource status data comes from a wide range of sources and formats
Data format is not uniform	The log, indicator, and configuration data structures differ greatly
Complex transmission link	It involves several intermediate nodes and conversion links
Real-time synchronization difficulty	The data update frequency is high, and the synchronization mechanism is easy to fail

As shown in Table 1, the more heterogeneous the data, the more preprocessing work is required for scheduling. The more complex the transmission path, the longer the system's transmission delay. As a result, the scheduling system cannot adapt to the resource status in time. If a standardized and low-delay data collection and transmission mechanism is not built, the scheduling algorithm will continue to rely on "old data". This will lead to declining resource use efficiency, adversely affecting scheduling stability and service quality.

3.2. Deviation of State Perception and Prediction: Weakens the Accuracy of Resource Allocation

Cloud platform high scheduling depends on the comprehensive judgement of the accuracy of current device status and future development load estimation. However, the status perception module of the system often experiences issues such as collection lag and insufficient monitoring, resulting in an inability to timely and accurately collect the current working status of nodes [5]. For example, the slow refresh of some parameters (memory utilization, I/O bottlenecks, etc.) can lead to inauthentic system decisions. And the current mainstream resource prediction mechanism mainly establishes static model to learn, which cannot adapt to the dynamic behavior and resource demand of users in the cloud environment. Common dependent variables that affect state perception and prediction bias are shown in Table 2:

Table 2. Typical Effects of State Perception and Prediction Bias.

Problem link	Concrete impact
Status acquisition delay	The perception of the scheduling system is distorted
Insufficient monitoring accuracy	Some load changes are not accurately captured
The generalization ability of the model is weak	The predicted result differs greatly from the actual resource demand
Lack of feedback mechanism	Lack of closed-loop data updates to support dynamic optimization

As shown in Table 2, when the state awareness rate is slow, the scheduling policy relies on "late information", which may lead to wrong decisions and misleading resource allocation when the prediction model's accuracy is low.

3.3. Policy Rigidity and Dynamic Disconnection: Reduced System Adaptability

Although the cloud platform still uses a static, rule-based scheduling mode, it has many defects in practical application. These scheduling policies, implemented with predefined parameters, lack sensitivity and timely response to environmental changes. Therefore, dynamic adjustment cannot be performed quickly when facing a large number of loads or unexpected requirements. For example, a sudden surge in workload or abnormal user operations often leads to the inability of this mode to dynamically shift to the optimal resource allocation strategy. This requires manual intervention or subsequent correction, which greatly affects the overall consistency of the service. The traditional scheduling strategy lacks frequent updates, a fixed scheduling process, and the ability to learn, preventing it from achieving self-evolution. Table 3 below summarizes the basic characteristics of policy rigidity and dynamic disconnection:

Table 3. The Key Manifestations of the Disconnection between Rigid and Dynamic Strategies.

Policy characteristics	Existing problem
Fixed resource allocation rules	It is difficult to match multiple task types
Lack of adaptive logic	The policy cannot be adjusted based on the real-time status
The adjustment period is too long	The adjustment mechanism relies on manual or static rules
Load abnormal response lags	Scheduling lags and efficiency decreases under peak load

As shown in Table 3, if the policy lacks flexibility and adaptability, it will negatively impact resource allocation, especially in cloud computing environments with high frequency of change and unpredictability. To ensure system adaptability, an intelligent scheduling strategy based on real-time insights, automatic adjustments, and rapid responses is developed to meet task needs and ensure timely responses.

3.4. Interface Coupling and Synchronization Imbalance Prevents Stable Execution Processes

The scheduling management module consists of multiple components that interact through interfaces, such as task scheduling, message feedback, and data sharing. If the interface design is not standardized and the communication protocol lacks unity, the efficiency of collaborative transmission between modules will decrease. This, in turn, increases the difficulty of designing and debugging each module. Most of the modules are related, so once a module fails, the overall scheduling work may stop. In addition, the time synchronization mechanisms used by each module vary, such as time label matching and state refresh intervals. This leads to task state confusion, multiple submissions, or preemption issues. Without a standardized time, synchronization and anomaly isolation scheme, the system's stability in complex environments will be significantly reduced. Resource scheduling may even fail. Table 4 below summarizes the bottlenecks of the system coupling interface and synchronization imbalance:

Table 4. System Bottlenecks of Interface Coupling and Synchronization Imbalance.

System bottleneck	Form of expression
Interface specifications are inconsistent	The communication cost between modules is high
Data synchronization modes are dispersed	Synchronization timing is inconsistent, causing scheduling failure
High dependence between modules	A module exception may affect the overall process
The exception handling mechanism is insufficient	Abnormal states lack real-time feedback and isolation mechanisms

As shown in Table 4, excessive interface coupling or synchronous control harms system modularity and execution flexibility. Errors in the control process can cause the scheduler to collapse.

4. Cloud Service Resource Allocation Efficiency Improvement Path Based on Data Engineering

4.1. Thin Data Architecture to Improve Scheduling Response Speed

The scheduling system depends heavily on the data architecture, and its recovery time is mainly determined by the architecture's complexity. Traditional data transmission processes involve multiple stages, such as intermediate conversion, repeated interpretation, and invalid storage, which delay the transmission of data to the scheduler and significantly prolong recovery time. The architecture design must be optimized and simplified by eliminating unnecessary transformations and packaging to reduce data processing time. By using an event-driven mechanism to acquire and process stream data, combined with edge computing technology, part of the computing logic is placed near the original data to shorten data feedback time. The data path is minimized, conversion logic is integrated, and parallel processing capabilities are enabled. Efficient, low-load channels tailored to scheduling requirements are built to further improve the system's overall resource recovery time.

To quantitatively describe the relationship between scheduling response rate and data transmission process, the following equation applies:

$$T_r = T_c + \sum_{i=1}^{n} (T_{ei} + T_{ti}) \quad (1)$$

Where T_r is the scheduling reaction time, T_c is the scheduling calculation time, T_{ei} is the time required for the data required for the i working step, and T_{ti} is the time required for the i transmission step. The goal of architectural modeling is to minimize the sum of the various T_{ei} and T_{ti} as much as possible, thereby minimizing T_r.

4.2. Introduction of Prediction Model to Enhance the Accuracy of Resource Matching

Traditional fixed scheduling rules struggle to adapt to the changing demand environment. Using artificial intelligence predictive models such as time series models and neural networks (e.g., LSTM, GRU), future resource demand can be accurately predicted.

These models can also learn workload arrival rates, resource occupancy patterns, historical load trends, and other factors, predicting potential system pressures to feed back to the scheduling engine for decision-making. Compared to fixed static scheduling strategies, the AI prediction model offers greater self-adjustment ability and adaptability. It can flexibly adjust the configuration when it needs to be adjusted, so as to alleviate the occurrence of resource waste and congestion. Meanwhile, the characteristics of continuous training and real-time update ensure the accuracy of its effect in long-term scheduling work. It aids in improving the intelligence level and matching rate of the scheduling system.

4.3. Improve the Perception Mechanism to Realize Dynamic Task Scheduling

To achieve dynamic scheduling, we need to sense and process the system's state, resource status, and task changes in real time. In traditional designs, perception is mainly achieved through periodic queries, which result in long response times and slow data updates, making dynamic task adjustment difficult. Therefore, improving sensing performance is essential. An event-driven real-time response mode based on status subscription should be implemented to timely acquire parameters such as CPU usage, memory load, and network traffic, enabling automatic transmission. At the same time, a diversified perception and comprehensive index evaluation method is used to detect sudden task or load increases. Based on these findings, the resource allocation strategy is reconfigured, and the scheduling policy engine is used to take preventive measures before resource shortages occur. For example, transferring tasks or adjusting priorities to ensure the normal operation of the system.

According to the above path, a dynamic response function of scheduling state can be constructed to represent the influence of key indicators on scheduling behavior:

$$S_d(t) = f\big(R_s(t), T_w(t), U_n(t)\big) \quad (2)$$

Where $S_d(t)$ is the change response of scheduling behavior at this time, and f is the association rule of sensing mode to resource $R_s(t)$, task waiting time $T_w(t)$ and node load $U_n(t)$ in the system. This will greatly optimize the sensitivity and accuracy of $S_d(t)$ to support the immediacy of scheduling tasks.

4.4. Strengthen Platform Coordination and Promote Overall Resource Optimization

Cloud computing often involves various platforms, clusters, or server nodes, with resources on each node being heterogeneous and separate. To maximize the utility of the entire system, a collaboration model across platforms is needed. Artificial intelligence can help achieve comprehensive coordination and regulation of network resources. By using methods such as multi-agent reinforcement learning or graphical neural networks, the interaction process between platforms can be simulated. Based on the results, corresponding scheduling and work plans are formed. This method considers not only the load of a single node but also parameters such as network status, task type, and priority, scheduling tasks and migrating resources across platforms for system-level optimization.

5. Conclusion

In order to improve the quality of resource scheduling of cloud services, we must rely on the technical support and intelligent decision-making functions of data engineering and artificial intelligence, optimize the data collection process, strengthen the system of model construction and management, and effectively resolve the current reaction lag and distribution imbalance of resource scheduling with the help of predictive algorithms and dynamic adjustment strategies. Through AI technology to achieve the understanding of the situation, forecast work needs and scheduling, has become an important means to establish an efficient resource allocation management system, but also to continuously improve the cloud service platform work efficiency and intelligence level has laid a solid foundation.

References

1. J. I. Siepmann and J. Sanders, "Announcing the J. Chem. Eng. Data Early Career Award," *J. Chem. Eng. Data*, vol. 68, no. 8, pp. 1833-1833, 2023, doi: 10.1021/acs.jced.3c00457.
2. A. Lekova, P. Tsvetkova, and A. Andreeva, "System software architecture for enhancing human-robot interaction by conversational AI," in *Proc. 2023 Int. Conf. Inf. Technol. (InfoTech)*, 2023, pp. 1-6, doi: 10.1109/InfoTech58664.2023.10266870.
3. Y. Liu, C. Liang, J. Wu, H. Jain, and D. Gu, "A group consensus decision-making method for cloud services selection with knowledge deficit by trust functions," *Kybernetes*, vol. 53, no. 1, pp. 337-357, 2024, doi: 10.1108/K-03-2022-0422.
4. O. Kozinski, M. Kotyrba, and E. Volna, "Improving the production efficiency based on algorithmization of the planning process," *Appl. Syst. Innov.*, vol. 6, no. 5, p. 77, 2023, doi: 10.3390/asi6050077.
5. D. Wen, X. Li, X. Ren, M. Ji, and Q. Long, "Optimisation of berth and quay crane joint scheduling considering efficiency and energy consumption," *Int. J. Shipp. Transp. Logist.*, vol. 17, no. 4, pp. 487-505, 2023, doi: 10.1504/IJSTL.2023.136048.

Disclaimer/Publisher's Note: The views, opinions, and data expressed in all publications are solely those of the individual author(s) and contributor(s) and do not necessarily reflect the views of PAP and/or the editor(s). PAP and/or the editor(s) disclaim any responsibility for any injury to individuals or damage to property arising from the ideas, methods, instructions, or products mentioned in the content.