GEORGE BROWN PRESS
JOURNALS, BOOKS & PROCEEDINGS

Journal of Computer, Signal, and System Research

Article

Intelligent Quality Governance Framework and Adaptive Verification Model in Cloud Integration Scenarios

Mingde Guo [1,*]

[1] Amazon, Irvine, CA, 92620, United States
* Correspondence: Mingde Guo, Amazon, Irvine, CA, 92620, United States

Abstract: The background of cloud integration is a huge system composed of numerous distributed services. The relationship between data flow and service invocation is constantly changing dynamically, thus making quality risks highly concealed and contagious. However, the traditional approach relies on manual analysis and static scanning to assess quality, which is difficult to meet the complex environmental requirements of various types, large quantities, high data density and frequent changes. For this purpose, this paper proposes the Intelligent Quality Governance Framework (IQGF), which establishes a representation system of quality measurement by integrating monitoring data, logs and call chains. At the same time, a self-regulating inspection model is proposed, which can automatically adjust the inspection range and plan according to the real-time risk information to achieve dynamic detection based on the operation status. This method shifts quality management from passive detection to active management, providing reliable quality assurance for large-scale cloud integration systems.

Keywords: cloud integration; quality governance; multi-source data; adaptive verification

Received: 10 November 2025
Revised: 25 November 2025
Accepted: 14 December 2025
Published: 20 December 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/).

1. Introduction

In cloud-integrated systems, the continuous growth in the number and diversity of services has led to increasingly frequent changes in system topology, operational load, and resource scheduling strategies. These dynamic adjustments, while essential for improving flexibility and scalability, significantly increase uncertainty in system behavior and pose substantial challenges to quality assurance. In particular, service instances are constantly created, migrated, or terminated, and resource allocations are adjusted in real time, making system states highly dynamic and difficult to predict using static assurance mechanisms [1].

Moreover, the data required for quality assurance are often distributed across multiple layers, including infrastructure, platform, and application levels. Due to the heterogeneity of data sources and the absence of unified association rules, it is difficult to establish effective correlations among operational metrics, service behaviors, and quality indicators. As a result, fault diagnosis and quality evaluation processes tend to rely on fragmented information, which weakens their accuracy, timeliness, and interpretability, especially in large-scale and highly coupled service environments.

Traditional verification and validation methods are generally designed for relatively stable system architectures and predefined execution paths. In cloud-integrated systems, however, frequent version iterations and complex service interactions make it difficult for these methods to respond promptly [2]. Verification activities are often conducted offline or at discrete time points, which limits their ability to capture transient faults and evolving performance degradation. Consequently, confirmation blind spots are likely to emerge during dynamic execution processes, reducing the effectiveness of quality assurance and increasing operational risks.

To address these challenges, it is essential to construct intelligent quality assurance technologies that can seamlessly integrate heterogeneous data, assurance policies, and confirmation procedures. Such technologies should enable continuous perception of system states, adaptive analysis of quality risks, and timely feedback during system execution. By supporting continuous monitoring, automated reasoning, and self-regulation mechanisms, cloud-integrated systems can dynamically adjust their behavior in response to detected anomalies or quality deviations. This capability is critical for meeting the stringent quality assurance requirements of large-scale integrated systems operating in highly dynamic and uncertain environments [3].

2. The Technical Mechanisms and Key Issues of Integrated Quality Governance

2.1. Engineering Characteristics of the Aggregated Environment

The cloud-integrated operation structure has a high degree of fluidity. The invocation of service entities depends on factors such as changes in workload, scheduling techniques, and elastic strategies. Therefore, their structural forms are always in a state of dynamic change, and the entire network topology is also continuously evolving. Furthermore, the same request may take completely different paths at different points in time. This uncertainty makes it difficult to precisely describe the system's behavior.

Distributed systems are simultaneously affected by multiple factors such as resource competition, network jitter, virtualization overhead, and node heterogeneity. As a result, the system will experience significant fluctuations in latency and throughput within a short period of time. Due to the strong interdependence among services, these local fluctuations are prone to be amplified in the link, which in turn leads to overall performance degradation or cross-node failures [4]. The data in the cloud aggregation environment presents multi-source heterogeneous characteristics, with different sources, granularities, semantics, and temporal accuracies. Monitoring indicators, link tracing, log events and experimental results and other information are often stored in various forms in a decentralized manner. The system behavior also has event-driven characteristics. Under the trigger of events such as update cycles, release frequencies, traffic peaks or sudden anomalies, it may undergo extensive state changes and exhibit obvious periodic disturbances and nonlinear responses.

2.2. Cloud Integration as the Main Challenge of Quality Governance

Due to the increasingly rapid transformation of the system structure, the transmission path of quality issues has become more complex. The vulnerable points of any node may spread downward along the dependency relationship, causing the problem manifestation to be disconnected from the root cause, thereby increasing the difficulty of problem detection. The lack of consistency in data from multiple sources is another major challenge in problem handling. Monitoring, logging and link tracking systems operate independently, with inconsistent data forms and time norms, making it difficult to construct a synchronous causal relationship and reducing the accuracy of quality inspection.

The existing verification methods are unable to cope with the dynamic operating state of the system. Services change frequently, network graphics keep evolving, and verification based on fixed scenarios is hard to cover all actual situations. Some latent risks are often only exposed during the production process. From a management perspective, there is a lack of an overall perspective. A single indicator cannot describe the overall effect of multiple services working together, and the quality impact relationship across services and levels has not been effectively expressed either.From the perspective of management processes, automatic loops are still lacking. Risk identification, strategy development and verification implementation all require human participation, with limited response capabilities, making it difficult for management efficiency to track system changes in a timely manner.

3. Design of the Intelligent Quality Governance Framework (IQGF)

3.1. Overall Framework Architecture

The design logic of IQGF is based on a comprehensive understanding of the operational characteristics of the cloud aggregation environment. Due to the continuous generation of various types of information and the constant changes in service structure, the system performance will constantly change along with the variations in resource allocation and workload. Therefore, this framework needs to have the ability to monitor the system status in real time, flexibly interpret the correlations between services, proactively identify the essence of problems, and provide the best solutions, among other functions.

The data flow expands from left to right: Data from various control sources of quality management first enters the data fusion unit, and after purification and feature extraction, it is transformed into arithmetic features. Subsequently, it enters the quality management network to endow it with intelligent processing capabilities and correlation analysis capabilities; Then, through the intelligent decision-making unit, on-site quality issues are predicted, and feedback is provided and corresponding management measures are implemented, thus forming a complete cycle process. The main components of the framework can be presented in the form of Figure 1.

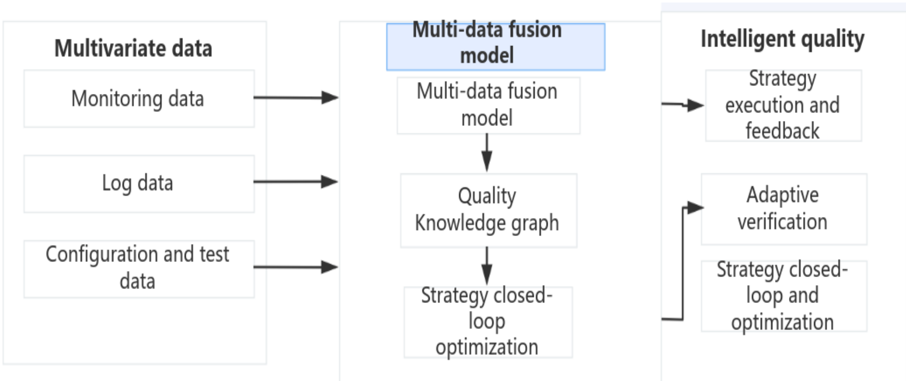

Figure 1. Schematic diagram of the overall structure of the intelligent quality governance framework.

3.2. Multi-source Data Fusion Model

Quality information entering the cloud integration system from various sources includes monitoring indicators, link tracing, log data, test results, etc. Due to the differences in types, semantics and temporal accuracy among various types of data, if they are not integrated.

If combined processing is carried out, it will be difficult to make an effective assessment of the quality status and it will also be impossible to form unified data in the same spaceSource. Therefore, in order to achieve the goal of unified data source modeling, IQGF has constructed multi-level quality information fusion Mode. The type characteristics and structures of different quality information are shown in Table 1.

Table 1. Structural Attributes and Data Characteristics of Multi-source Quality Data.

Data type number	Data scale (entries per day)	Time Granularity (ms	Structural complexity (1-5)	Noise ratio (%)
1	10^6	100-1000	2	5-10
2	10^5-10^6	<1	4	1-3
3	10^7	Event-driven	5	15-30
4	10^2-10^3	Not fixed	1	0-2

5	10-100	Second level	1	0

The fusion model first maps various types of data into feature vectors, providing a foundation for subsequent calculations. Fusion can be represented by linear combinations:

$$F = \alpha M + \beta L + \gamma T \quad (1)$$

Among them, M is the monitoring feature vector; L is the log feature vector; T is the link feature vector; α, β, γ is the weight. For this purpose, the errors of each feature element in each time series were reduced, and the feature vectors were homogenized:

$$\hat{F}_i = \frac{F_i - \min(F)}{\max(F) - \min(F)} \quad (2)$$

Among them, F_i is the i-th original feature value in the multi-source fusion feature vector; min(F) is the minimum value in the feature vector F; max(F) is the maximum value in the eigenvector F; $\hat{F}_i$ is the normalized i-th eigenvalue. The normalized features can participate in the calculation in collaboration with data from different sources, making the identification of quality issues more consistent and reliable.

3.3. Construction of Quality Knowledge Graph

In a cloud-integrated environment, service dependencies are complex, and a single data point is difficult to express the impact of metrics on quality. IQGF builds a quality knowledge graph to graphically represent elements such as services, interfaces, metrics, and events for better management.The source of quality change and its influence path. In a knowledge graph, nodes represent entities of different qualities: services represent business components, interfaces represent calling relationships, metrics represent operational status, and events represent abnormal behaviors. Edges are used to describe the associations among calls, metrics, and events. For details, see Table 2.

Table 2. Data Attribute Table of Entities and Relationships in the Quality Knowledge Graph.

Entity type number	Relationship type number	Relationship strength (0-1)	Dependency Levels (1-5)	Event frequency (times per day)	Information weight (1-5)	Node activity (0-1)
1	1	0.30-0.90	3-5	100-500	5	0.60-0.95
2	2	0.20-0.80	2-4	200-1000	4	0.50-0.90
3	3	0.10-0.70	1-3	10^4-10^6	3	0.40-0.85
4	4	0.10-0.60	1-2	10^3-10^5	2	0.30-0.80
5	5	0.40-1.00	1-2	1-20	5	0.20-0.60

In a knowledge graph, the degree of influence of each relationship can be expressed through weights. The weight calculation adopts the combined factor model:

$$w_{ij} = \lambda_1 f_{lat} + \lambda_2 f_{err} + \lambda_3 f_{freq} \quad (3)$$

Among them, f_{lat} is the delay impact factor; f_{err} is the error influencing factor; f_{freq} is the call frequency factor; w_{ij} is the quality correlation weight between node i and node j. λ_1,λ_2 and λ_3 are the weight coefficients of the three influencing factors. The weights are dynamically updated along with the changes in the operating state, keeping the knowledge graph sensitive to the real-time status. Knowledge graphs enable governance engines to no longer rely on a single metric for judgment, but to understand quality issues based on system-level dependency structures, providing structured basis for strategy generation.

3.4. Intelligent Quality Decision Engine

The decision engine is at the core of IQGF. The fused feature vectors and graph structure are input into the risk assessment model to generate a real-time score of the system quality.

$$R = \sum_{i=1}^{n} w_i \hat{F}_i \quad (4)$$

Among them, F_i represents the normalized feature; w_i represents the feature importance weight; R represents the overall quality risk score of the system; n is the dimension of the fusion feature vector; $\hat{F}_i$ is the normalized value of the i-th feature. When the score exceeds the threshold, the decision engine will trigger the corresponding governance operation. Governance strategies include supplementary verification, traffic dilution, link stress testing, service degradation or dependency adjustment, etc. Dynamic strategies are chosen based on risk sources and their impact scope, with execution results fed back to update the model. Through data-driven analysis and knowledge inference, IQGF identifies risks and generates adaptive strategies, enabling a closed-loop intelligent governance process.

4. Adaptive Verification Model (AVM) Method System

4.1. Adaptive Verification Trigger Mechanism

The triggering of adaptive verification does not initiate the self-correction process based on pre-established rules, but is dynamically activated during the system's operation, such as increased latency during operation, repeated faults, network fluctuations, topological changes, fluctuations in available resource levels, and significant changes in relationship weights in the quality knowledge graph. The key lies in seizing the "potential danger signals" rather than dealing with the problem only when it has deteriorated to the extent that it affects the user experience.

Commonly used trigger signals can be classified into three categories: operational state fluctuation signals, structural change signals, and abnormal event signals. Operational state fluctuations may imply changes in latency or a decline in local capabilities; Structural changes may be manifested in aspects such as path migration, instance replacement, or hierarchical structure changes; Abnormal events include a large number of alerts in logs and abnormal closed paths in tracking information, etc. AVM determines whether verification is needed based on the degree of influence, the rate of change, and the range of change of the signal to avoid unnecessary large-scale testing investment. The advantage of the trigger mechanism is that it can shorten the extension time after the appearance of risk factors. When there are minor abnormalities in system stability, remedial checks, critical path checks or extended checks can be initiated immediately without relying on manual intervention, thereby improving the efficiency of problem capture and reducing the risk of service quality decline.

4.2. Verify Task Classification

Adaptive verification should select self-adjusting tasks based on the execution state. The decomposition method needs to reflect the behavioral variations caused by various potential risk factors that may occur in the execution state. For example, potential risk factors in a cloud aggregation environment include network accessibility performance, device performance, interface recovery, abnormal scenarios, and data crossover phenomena, etc. For this, the test tasks can be decomposed into tasks such as network connectivity detection, service quality detection, interface recovery detection, abnormal situation simulation testing, and data synchronization detection.

The degree of correlation between different tasks and between tasks and signals varies. During operation, the probability of each task being activated will be constantly adjusted as the risk status changes. To reflect this dynamic selection relationship, a task selection probability model can be established by performing weighted calculations on different verification tasks.

$$T_k=\frac{\omega_k \cdot S_k}{\sum_{j=1}^{m} \omega_j \cdot S_j} \tag{5}$$

Among them, T_k is the trigger probability of the KTH type task; ω_k represents the fundamental weight of the task, reflecting the importance of the task itself. S_k represents the intensity of the risk signal related to this task in the current operating state. m represents the total number of task categories; $\omega_j S_j$ is the sum of the weight-signal fusion values for all task categories.

4.3. Adaptive Verification of Model Structure

The adaptive verification model is composed of modules such as operational status perception, target extraction, path inference, and task scheduling. It is not a fixed process but dynamically adjusts according to the system status to adapt to changes in risk signals and dependencies. The operational state awareness module characterizes the system state based on information such as delay jitter, error rate density, and link drift, and extracts risk vectors as the basis for strategy selection. To quantify the immediate state, a state fusion model can be adopted to obtain a comprehensive representation:

$$H=f(F_1,F_2,\ldots,F_n) \quad (6)$$

Among them, H represents the characterization of the operating state of the fused system; F_i represents the I-th type of operational state characteristics (such as delay changes, error events, or link shifts); f(.) As a fusion function, it can be implemented in a linear or weighted manner. After the operational state characteristics are obtained, the model enters the structural inference stage. At this stage, key nodes are identified by combining knowledge graphs, dependency weights and risk vectors, and candidate verification paths are generated. The paths are scored based on the intensity of dependence and the possibility of risk transmission, making the verification more concentrated in high-risk areas. Path scoring can be described by the following function:

$$S_p=\sum_{(i,j)p} \alpha\, w_{ij}+\beta R_i \quad (7)$$

Among them, S_p is the comprehensive score of path p; w_{ij} is the weight of the dependency relationship; R_i is the risk value of node i; α and β are the regulating parameters. After the path is determined, the scheduling module will dynamically schedule based on the path score, system load and verification cost, and send the result to the controller to arrange the execution sequence, so that high-risk paths are prioritized for processing, while the frequency of low-priority tasks is reduced when the load is high.

4.4. Verify the Strategy Adaptive Algorithm

The core of AVM adaptation is implemented through policy algorithms. This algorithm enables the verification action to adapt to the dynamic operating environment by selecting the verification object, expanding the verification path and adjusting the verification intensity. The selection of verification objects is jointly determined by risk signals, dependencies, and the influence of nodes. The selection process is calculated through a priority scoring model and expressed as follows:

$$P_i=\theta_1 R_i+\theta_2 w_i+\theta_3 d_i \quad (8)$$

Among them, P_i represents the verification priority of the node; R_i stands for Node Risk score; w_i represents the influence weight of the node in the knowledge graph; d_i stands for dependency depth; θ_1,θ_2, and θ_3 are the adjustment coefficients. This model enables nodes with significant risks, complex dependencies or wide influence to automatically enter the priority verification sequence.

Verify the extended dependency chain weights of the path. The extended logic follows the principle of "high association relationship priority", and the expression is as follows:

$$E(S)=\{j|w_{ij}>\tau,\ i\in S\} \quad (9)$$

Among them, S represents the current set of verification targets; w_{ij} is the quality correlation weight between nodes; τ is the extended threshold; j represents the "adjacent nodes" that may be included in the extended verification path; E(S) is a new set of nodes obtained by extending the set S. This function ensures that the verification path only extends to nodes that truly have the possibility of risk propagation. The strength of the system is determined based on the measurement of system load, failure rate and delay fluctuation, in order to achieve real-time management of test strength. Under high test pressure, the system will reduce the workload or lower the test frequency. When the high-risk status is relatively high, it is possible to increase the test depth, expand the range of the test path, or adopt a parallel testing approach.

5. Conclusion

In the face of the complexity of integrated environmental quality governance, the constructed intelligent governance system can still maintain reliable quality assessment performance in an environment with dynamic interconnection structures, different types of data pollution, and high-frequency changes. The combined application of data fusion, knowledge representation and adaptive confirmation methods can quickly identify the main affected areas after abnormal situations are discovered and automatically carry out corresponding confirmation activities, thereby improving the quality and efficiency of governance. Intelligent quality management technology can be further expanded on this basis to be used for modeling methods in the training process, anomaly identification capabilities, and consistency handling among different cloud platforms. Introducing adaptive confirmation methods into maintenance automation and artificial intelligence analysis and other links is expected to form a comprehensive autonomous quality management process, thereby better supporting the stable operation of large-scale cloud integration systems.

References

1. E. Q. Shahra, W. Wu, S. Basurra, and A. Aneiba, "Intelligent edge-cloud framework for water quality monitoring in water distribution system," *Water*, vol. 16, no. 2, p. 196, 2024. doi: 10.3390/w16020196
2. Y. A. N. G. Xian-Min, Z. E. N. G. Jia-Yao, Z. H. A. O. Lu-Yao, and L. I. Xin, "Design of postgraduate high-quality teaching resource sharing mechanism and implementation path from the perspective of community theory," *Modern Educational Technology*, vol. 34, no. 10, pp. 75-82, 2024.
3. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 581-585, 2019.
4. C. Ren, X. Wen, X. Guan, C. Chen, and Y. Ma, "AIoT for aircraft final assembly: An intelligent and collaborative framework," *Journal of Communications and Information Networks*, vol. 9, no. 3, pp. 262-276, 2024. doi: 10.23919/jcin.2024.10707102

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of GBP and/or the editor(s). GBP and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.