Cloud-Based Broadcasting Converged Media Data Processing Architecture Design: Postprint

Authors: Chen Hao

Date: 2025-07-09T15:34:21+00:00

Abstract

【Objective】To address the challenge that traditional data processing architectures cannot satisfy business requirements under new circumstances—given that broadcast media convergence data exhibits multi-source heterogeneous, highly real-time, and large-scale characteristics—a cloud computing-based data processing architecture is designed.

【Method】Employing distributed storage and computing technologies, a multi-layer architecture system comprising data acquisition layer, storage layer, computing layer, and service layer is constructed; business decoupling is realized through microservices architecture, container technology is leveraged to enhance system elastic scaling capability, and a stream computing engine is introduced to handle real-time data streams.

【Results】System testing shows that this architecture improves data processing efficiency, system throughput, and response time by over 50% compared with traditional architectures, while reducing node scaling time by 80%.

【Conclusion】The cloud computing-based broadcast media convergence data processing architecture demonstrates strong practical value and can provide effective technical support for the deep convergence development of broadcast media.

Full Text

Design of a Cloud Computing-Based Data Processing Architecture for Radio and Television Converged Media

Rong Media Center of Pingyi County, Linyi City, Shandong Province, Linyi, Shandong 273300, China

Abstract

[**Purpose**] To address the challenges posed by the multi-source heterogeneous, real-time, and large-scale characteristics of radio and television converged media data, which render traditional processing architectures inadequate for emerging business requirements, this paper proposes a cloud computing-based data processing architecture. [**Method**] The architecture employs distributed storage and computing technologies to construct a multi-layer system comprising data acquisition, storage, computing, and service layers. Business decoupling is achieved through microservices architecture, container technology enhances system elasticity and scalability, and a stream computing engine is introduced for real-time data flow processing. [**Results**] System testing demonstrates that compared with traditional architectures, the proposed architecture improves data processing efficiency, system throughput, and response time by over 50%, while reducing node scaling time by 80%. **Conclusion** The cloud computing-based radio and television converged media data processing architecture offers significant practical value and provides effective technical support for the deep integration and development of broadcast media.

Keywords: Cloud Computing; Radio and Television Converged Media; Data Processing; Wireless Connection Device; Distributed Architecture; Microservices
CLC Number: G222
Document Code: A
Article ID: 1671-0134(2025)05-90-04
DOI: 10.19483/j.cnki.11-4653/n.2025.05.019
Citation Format: Chen H. Design of a Cloud Computing-Based Data Processing Architecture for Radio and Television Converged Media [J]. China Media Technology, 2025, 32(5).

Introduction

The development of radio and television converged media represents a crucial initiative for promoting the deep integration of traditional and emerging media. In the converged media environment, broadcast institutions must simultaneously process multiple data types including traditional radio and television programs, new media content, and user interaction data, imposing higher demands on data processing systems. Particularly in wireless data acquisition and transmission, efficient and reliable wireless connection devices must be designed to ensure real-time and accurate data collection. Cloud computing, with its distributed and elastically scalable characteristics, combined with dedicated wireless connection devices, offers new solutions for addressing data processing challenges in radio and television converged media. The key question of how to construct an efficient and reliable data processing architecture based on cloud computing and wireless connection technologies has become an urgent issue in broadcast media convergence development.

Radio and television converged media data exhibit characteristics of multi-source heterogeneity, massive volume, and high timeliness, encompassing broadcast television, social media, and user behavior data [1]. Traditional data processing architectures face storage bottlenecks, limited throughput, and expansion difficulties, making them unable to meet rapid business development needs. With the maturation of distributed computing, containerization, and edge computing technologies, coupled with the resource elasticity and on-demand allocation features of cloud computing platforms, the technical foundation has been established for building a new-generation converged media data processing system.

The system architecture adheres to four core principles: high availability, scalability, high performance, and security. High availability is achieved through distributed cluster deployment and multi-replica data backup to ensure 24/7 stable operation. Scalability relies on microservices architecture to enable horizontal and vertical system expansion. High performance is realized through distributed computing for parallel data processing and load balancing. Security employs a multi-layered protection system including data access control and transmission encryption mechanisms [2]. The system adopts a layered design with standard interfaces between layers to achieve loose coupling, incorporating cloud-native features such as elastic scaling and fault isolation.

2. System Architecture Design

2.1 Architecture Design Principles

The system architecture follows four fundamental principles: high availability, scalability, high performance, and security. High availability ensures 7$×$24-hour stable operation through distributed cluster deployment and multi-replica data backup. Scalability is achieved via microservices architecture enabling both horizontal and vertical system expansion. High performance leverages distributed computing for parallel data processing and load balancing. Security implements a multi-layered protection system encompassing data access control and transmission encryption mechanisms [2]. The layered design facilitates loose coupling between components through standard interfaces while incorporating cloud-native characteristics such as elastic scaling and fault isolation.

2.2 System Function Module Division

The system is divided into four primary functional layers: data acquisition layer, distributed storage layer, distributed computing layer, and microservices layer (see [Figure 1: see original paper]). The data acquisition layer handles unified access to multi-source heterogeneous data, including wireless signal acquisition modules, social media data acquisition modules, user behavior data acquisition modules, and video stream data acquisition modules. The distributed storage layer employs a hybrid storage architecture, selecting appropriate storage solutions for different data types: object storage for large-scale multimedia resources such as videos and images; distributed database clusters for structured data; and distributed file systems for semi-structured data. The distributed computing layer builds a unified data processing platform integrating stream computing engines, batch processing frameworks, real-time analysis engines, and data mining modules to enable both real-time processing and offline analysis. The microservices layer provides standardized interface services for applications, including content distribution services, data analysis services, user profiling services, and intelligent recommendation services. A comprehensive monitoring and early warning mechanism is designed between layers, equipped with performance metric collection probes and log analysis systems to visualize system operation status and ensure coordinated functioning of all modules [3].

2.3 Wireless Data Acquisition and Transmission Design

The wireless data acquisition and transmission module serves as the system data entry point, employing a distributed acquisition node deployment strategy to ensure comprehensive and reliable signal acquisition [4]. Acquisition node hardware utilizes high-performance Digital Signal Processors (DSP) and Field Programmable Gate Arrays (FPGA) to enable real-time signal sampling and preprocessing. At the transmission level, dedicated data transmission channels are established with two-way authentication mechanisms to ensure data transmission security. To improve transmission efficiency, data compression algorithms are introduced to achieve lossless compression, significantly reducing network bandwidth consumption. Additionally, a complete data caching mechanism is designed to ensure zero data loss during network fluctuations. Acquisition nodes feature intelligent scheduling capabilities, automatically adjusting sampling parameters based on signal quality to adapt to complex transmission environments. The module also integrates a signal quality monitoring system for real-time monitoring of signal metrics to provide decision support for system operations and maintenance [5]. Lightweight artificial intelligence algorithms are deployed at the edge to perform preliminary data cleaning and classification, achieving data governance at the source and improving subsequent processing efficiency.

3. Wireless Connection Device Design

3.1 Hardware Structure

The hardware structure of the wireless connection device adopts a modular design concept. As shown in [Figure 2: see original paper], it consists of core modules including an antenna system, RF front-end, signal processing unit, analog-to-digital conversion unit, main controller, power management system, and network interface. The antenna system employs a high-gain directional antenna array with beamforming technology to improve signal reception quality. The RF front-end integrates low-noise amplifiers and power amplifiers to achieve signal amplification and attenuation control. The signal processing unit adopts a DSP and FPGA hybrid architecture responsible for signal preprocessing and complex algorithm computation. The main controller runs a real-time operating system to coordinate management of all functional modules. The power management system provides multi-channel regulated outputs with protection functions. The network interface supports gigabit Ethernet and fiber optic access to ensure high-speed data transmission. The hardware design reduces internal interference through multi-layer PCB layout and shielding design.

3.2 Signal Acquisition and Processing Module

The signal acquisition and processing module is responsible for wireless signal reception, conditioning, and processing. This module employs broadband direct sampling technology, implementing adaptive signal conditioning through multi-stage amplification and filtering architecture. Digital down-conversion uses digital intermediate frequency technology to reduce analog circuit complexity. The FPGA is responsible for signal processing algorithm implementation, supporting multi-channel parallel processing, while the DSP performs spectrum analysis and signal demodulation using a pipeline architecture to improve efficiency. The module is equipped with an intelligent threshold detection algorithm that automatically identifies signal mutations and triggers data capture, ensuring real-time processing performance through high-speed caching and serial communication interfaces.

3.3 Data Transmission and Synchronization Mechanism

The data transmission and synchronization mechanism adopts a layered design strategy to construct a reliable data transmission network. The transport layer employs an improved TCP/IP protocol stack to achieve reliable data transmission and flow control. The system integrates the IEEE 1588 Precision Time Protocol (PTP) to achieve inter-system synchronization accuracy better than 100ns. Data transmission supports multiplexing technology, enabling simultaneous transmission of control signaling and business data. To improve transmission efficiency, an adaptive frame length control algorithm is designed to dynamically adjust data frame size based on channel quality. Transmission channels adopt AES-256 encryption algorithms to ensure data transmission security. The system also implements a sliding window-based flow control mechanism to effectively prevent network congestion. Data synchronization employs bidirectional timestamp technology to accurately record data acquisition and processing delays. The cache management module adopts a multi-level cache structure to reduce access latency through prefetch mechanisms. The system is equipped with a network status monitoring unit for real-time link quality monitoring and supports automatic switching to backup links to ensure transmission reliability.

3.4 Anti-Interference Design

The anti-interference design integrates multiple advanced technologies to establish a comprehensive anti-interference protection system. The RF front-end employs high-linearity amplifiers to improve system dynamic range and enhance suppression capabilities against strong interference signals. Signal processing adopts adaptive filtering algorithms that effectively suppress narrowband and pulse interference. The system integrates digital pre-distortion technology to reduce system nonlinear distortion through real-time monitoring and compensation. Spatial filtering employs adaptive beamforming technology to achieve spatial suppression of interference sources by adjusting the antenna array pattern. Time-domain processing introduces wavelet transform technology to improve detection capabilities for transient interference. Frequency-domain processing employs multi-resolution spectrum analysis to accurately identify interference characteristics. The system also establishes an interference source database and combines machine learning algorithms to achieve intelligent interference identification and classification. For EMC design, multi-layer shielding structures and special grounding techniques are adopted to significantly improve equipment electromagnetic compatibility performance.

4. Core Technology Implementation

4.1 Distributed Storage Solution

The distributed storage system adopts a multi-level hybrid architecture to build a unified storage platform for heterogeneous data. In the underlying storage architecture, data is categorized into three core components based on characteristics: object storage, distributed file systems, and time-series databases. Object storage employs the Ceph distributed storage system, ensuring data reliability through data sharding and multi-replication mechanisms, and enabling efficient storage and access of petabyte-level media resources. The distributed file system is built on HDFS using a master-slave architecture, with NameNode managing metadata and DataNode responsible for data storage, supporting large-scale unstructured data storage. For structured data generated by business systems, a distributed NewSQL database cluster is deployed using a sharded cluster mode to achieve horizontal data scaling. The system introduces a distributed caching layer using Redis cluster deployment with master-slave replication and sentinel mechanisms to ensure high availability and significantly improve hot data access efficiency. The storage system implements a unified data access interface, enabling global data indexing and cross-storage system data retrieval through metadata services. A tiered storage strategy is adopted, allocating hot data to memory, warm data to solid-state drives, and cold data to object storage based on data access characteristics to achieve optimal storage resource allocation [6].

4.2 Real-Time Computing Framework

The real-time computing framework builds a stream processing engine based on Apache Flink, combined with Apache Spark to achieve a unified computing platform for batch and stream processing. The computing framework adopts a distributed architecture design, with JobManager responsible for task scheduling and resource allocation, and TaskManager executing specific computing tasks. The system introduces Kafka message queues as a data buffering layer, using topic partitioning mechanisms to achieve parallel data processing. The computing framework supports core features including window computing, state management, and event-time processing, ensuring data processing consistency through a CheckPoint mechanism. For resource scheduling, the Kubernetes container orchestration platform is employed to achieve elastic scaling and fault self-healing of computing resources. The task scheduling system is built on a DAG workflow engine, supporting task orchestration and scheduling with complex dependency relationships. The computing framework integrates a machine learning algorithm library to provide intelligent analysis and prediction capabilities. The system implements a SQL-based query interface to lower development barriers and improve data analysis efficiency. Through an optimizer, automatic optimization of computing tasks is achieved, including operator reordering and resource allocation optimization strategies, significantly improving computing performance [7]. As shown in [Figure 2: see original paper], the real-time computing framework constructs a complete data processing pipeline to enable real-time data analysis and decision support.

4.3 Microservice Architecture Design

The microservice architecture adopts a domain-driven design approach, dividing system functions into core microservices such as content management, user analysis, and recommendation engines. The service governance layer employs the Spring Cloud framework to implement foundational functions including service registration, discovery, and configuration management. The service gateway is built on Kong, providing capabilities for routing, load balancing, rate limiting, and circuit breaking. Inter-service communication uses the gRPC protocol with Protocol Buffers for efficient data serialization. For distributed transaction issues, the Saga pattern is adopted to ensure data consistency. Service fault tolerance employs Hystrix for circuit breaking and degradation, preventing service cascading failures through thread pool isolation. The configuration center uses Apollo for unified configuration management and dynamic refresh. Service authentication is based on the OAuth2.0 protocol combined with JWT for stateless authentication. Microservice development follows the contract-first principle, with Swagger used for automated API documentation generation and management. The system adopts DevOps practices, building continuous integration and continuous deployment pipelines to improve service delivery efficiency [8].

4.4 System Monitoring and Operations

The system monitoring and operations platform constructs a comprehensive monitoring system for real-time system status monitoring and early warning [9]. The monitoring system uses Prometheus time-series database for storing monitoring metrics and Grafana for visualization. For application performance monitoring, SkyWalking is deployed for distributed tracing and call chain analysis to locate performance bottlenecks. Log management adopts the ELK architecture for centralized log collection and analysis, supporting complex queries and alert configuration. For resource monitoring, cAdvisor collects container resource metrics combined with Node Exporter for host status monitoring. The alerting system is built on AlertManager, supporting multi-channel alert push and alert rule configuration. For operations automation, Ansible is used for configuration management and automated application deployment, with Jenkins building continuous delivery pipelines. System security operations employ honeypot technology and deploy intrusion detection systems for real-time security threat monitoring and protection. The operations platform provides a unified operations portal integrating monitoring, alerting, deployment, and maintenance functions to improve operational efficiency [10].

5. System Testing and Performance Evaluation

5.1 Test Environment and Methodology

The test environment employs a distributed cluster architecture deploying 30 high-performance server nodes, each configured with Intel Xeon Gold 6248R processors, 256GB DDR4 memory, and NVMe solid-state storage [11]. The network environment uses 10 Gigabit Ethernet interconnects with SDN technology for intelligent network traffic scheduling [12]. The test dataset includes 300TB of video materials, 5 million user behavior records, and 1 million social media data entries, realistically simulating radio and television converged media business scenarios [13]. The test methodology follows a layered testing strategy covering unit testing, integration testing, performance testing, and stress testing. A distributed pressure testing platform is built using JMeter to simulate 50,000 concurrent user accesses with a test duration of 168 hours. For data processing performance, test cases are designed for typical business scenarios including video transcoding, real-time stream processing, and data retrieval. APM tools are used for full-link performance monitoring with ELK platform for log collection and analysis. Fault injection technology is introduced during testing to evaluate system fault tolerance and recovery mechanisms. Test data is automatically collected through the monitoring system to gather core metrics including CPU utilization, memory usage, and I/O throughput [11].

5.2 Performance Metrics Evaluation

System performance evaluation focuses on three core dimensions: throughput, response time, and resource utilization. For data processing throughput, video transcoding capability reaches 1,000 concurrent 4K resolution streams, and real-time stream processing throughput exceeds 1 million records per second. Data retrieval performance tests show that complex query average response time is controlled within 200ms [14]. System average response time remains below 50ms under high concurrency scenarios, with 95% of requests responding within 100ms. Resource utilization monitoring shows that under peak load, CPU average utilization maintains at 65%, memory usage stabilizes at 75%, and storage I/O utilization is approximately 60% [15]. Horizontal scaling tests verify the system's linear scalability, with overall system throughput improving nearly threefold after node expansion. Load balancing test results show balanced load distribution across nodes, with inter-node load differences controlled within 15%.

Conclusion

The cloud computing-based radio and television converged media data processing architecture achieves efficient processing of massive heterogeneous data through layered design and modular construction. This architecture fully leverages cloud computing technology advantages while providing excellent system scalability and operational convenience. Practice demonstrates that this architectural design can effectively support the business development needs of radio and television converged media and holds important reference value for promoting technological innovation in media convergence. Future work should further optimize data processing algorithms, enhance system intelligence levels, and strengthen data security protection mechanisms.

References

[1] Xu Xiaochun. Current Status and Application of Radio and Television Big Data in the Converged Media Era [J]. Information and Computer (Theoretical Edition), 2020(23): 28-29.
[2] Wu Honghui. Exploration of Architecture and Key Technologies for Radio and Television Station Converged Media Platform Construction [J]. Research on Communication Power, 2018(25): 248.
[3] Zhai Guangda. Research on Practice of Radio and Television Converged Media Service Cloud Platform Based on Hyper-Converged Architecture [J]. China Media Technology, 2021(12): 149-151.
[4] Li Zhuhong, Zhao Canming, Ji Shihou, et al. Development and Application of a Wireless Console Connection Device Based on Bluetooth Transmission [J]. Digital Technology and Application, 2017(9): 78-79.
[5] Li Zhihui, Chen Qian, Long Ling, et al. Design of a High-Definition Mirror Wireless Transmission Endoscope Device for Cattle and Sheep [J]. Guizhou Animal Husbandry and Veterinary Medicine, 2024(4): 37-38.
[6] Liu Xiaopeng. Analysis of Data Asset Accounting Treatment for Radio and Television Network Enterprises [J]. New Accounting, 2024(10): 56-60.
[7] Li Fang, Tao Deli. Cloud Computing Empowers Automotive Big Data Processing and Design of University Social Practice Platform [J]. Automotive Pictorial, 2024(11): 263-265.
[8] Yang Haolin, Hong Xian. Promoting "Smart Radio and Television +" to Create a "Hainan Model" for Media Convergence—A Case Study of Hainan Radio and Television Converged Media Center Development [J]. China Media Technology, 2023(9): 15-19.
[9] Interview with Yin Xunyu: China Radio and Television Converged Media Cloud Enables Traditional Radio and Television Professionals to Create

Programs with Ease [J]. TV Guide, 2019(2): 52-53.
[10] Wang Hongsheng, Xu Chao, Zhang Weidong. Discussion on Testing Methods for Radio and Television Converged Media Broadcasting Platform Based on Cloud Platform [J]. Radio and Television Information, 2020(5): 29-33.
[11] Wang Lei. Design of Security System for Radio and Television Converged Media Cloud Data Center [J]. Satellite TV and Broadband Multimedia, 2020(13): 208-209.
[12] Zhu Qian. Innovating Converged Media Monetization Models to Build a New Business Development Pattern for Municipal Radio and Television Media Groups [J]. China Media Technology, 2023(1): 123-126.
[13] Chen Feng. Research on Application of NIST Cloud Security Architecture in Radio and Television Converged Media Cloud [J]. Radio and Television Technology, 2021(8): 137-139.
[14] Cai Hongwu, Zhao Yajiao. Analysis and Practice of Security System Construction Scheme for Radio and Television Converged Media Cloud Platform [J]. Radio and Television Information, 2018(10): 111-115.
[15] Bai Hailiang. Analysis of Radio and Television Converged Media System [J]. West China Broadcasting TV, 2020(11): 184-185.

Author Information

Chen Hao (1972—), male, Han ethnicity, from Pingyi, Shandong, bachelor's degree, senior engineer. Research interests: radio and television converged media engineering technology and digital networks.

Responsible Editor: Li Yansong

Note: Figure translations are in progress. See original paper for figures.